

The Rh-1 full-size humanoid robot: Design, walking pattern generation and control

M. Arbulú*, D. Kaynov, L. Cabas and C. Balaguer

Systems Engineering and Automation Department, Robotics Lab, Carlos III of Madrid University, Spain

(Received 14 June 2008; final version received 15 July 2009)

This paper is an overview of the humanoid robot Rh-1, the second phase of the Rh project, which was launched by the Robotics Lab at the Carlos III University of Madrid in 2002. The robot mechanical design includes the specifications development in order to construct a platform, which is capable of stable biped walking. At first, the robots' weights were calculated in order to obtain the inverse dynamics and to select the actuators. After that, mechanical specifications were introduced in order to verify the robot's structural behaviour with different experimental gaits. In addition, an important aspect is the joints design when their axes are crossed, which is called 'Joints of Rectangular Axes' (JRA). The problem with these joints is obtaining two or more degrees of freedom (DOF) in small space. The construction of a humanoid robot also includes the design of hardware and software architectures. The main advantage of the proposed hardware and software architectures is the use of standardised solutions frequently used in the automation industry and commercially available hardware components. It provides scalability, modularity and application of standardised interfaces and brings the design of the complex control system of the humanoid robot out of a closed laboratory to industry. Stable walking is the most essential ability for the humanoid robot. The three dimensional Linear Inverted Pendulum Model (3D-LIPM) and the Cart-table models had been used in order to achieve natural and dynamic biped walking. Humanoid dynamics is widely simplified by concentrating its mass in the centre of gravity (COG) and moving it following the natural inverted pendulum laws (3D-LIPM) or by controlling the cart motion (Cart-table model). An offline-calculated motion pattern does not guarantee the walking stability of the humanoid robot. Control architecture for the dynamic humanoid robot walking was developed, which is able to make online modifications of the motion patterns in order to adjust it to the continuously changing environment. Experimental results concerning biped locomotion of the Rh-1 humanoid robot are presented and discussed.

Keywords: humanoid robot; mechanical design; software and hardware architecture; biped locomotion; inverted pendulum model; walking patterns; control system

I. Introduction

Since industrial robots cannot be easily adapted to assist human activities in everyday environments such as in hospitals, homes, offices, there is a growing need for robots that can interact with a person in a human-like manner. Wheel robots sometimes cannot be used in such types of environments because of the obvious restrictions posed by the use of wheels. For example, it is impossible for this kind of robot to go downstairs and upstairs or to clear some obstacles on the floor. What is more, humanoid robots are expected to play a more important role in the future.

One of the most exciting challenges that has faced the engineering community in recent decades was obtaining a machine of similar form, a humanoid robot, that could do the same activities as a human being in addition to walking in the same manner (such as HONDA robots, Hirai et al. 1998; HRP robots, Kaneko et al. 2002, 2008; Johnnie, Loeffler et al. 2003, 2004; LOLA, Lohmeier et al. 2006).

There are several reasons to construct a robot with these characteristics. Humanoid robots will work in a human atmosphere with greater effectiveness than any other types of

robots because the great majority of environments where they would interact reciprocally with a human are constructed taking into account the dimensions of the latter. If it is supposed that a machine should complete dangerous tasks or work in extreme conditions, in the ideal case its anthropometric measures must be as close as possible to the ones of its prototype. Inclusively, there are professionals who adduce that for a human being to interact naturally with a machine, it must look like him.

The main goal of this project is the development of a reduced weight human-size robot which can be a reliable humanoid platform for implementing different control algorithms, human interaction, etc.

The main assumption for the mechanical design started with the weight of a 1.20 m person and the desired walking motion of the humanoid robot. With these, the requirements for each joint's torque were calculated and then, by dynamical analysis, the structure was designed and the dimensions of the motors were determined. It was an iterative process for obtaining the optimal torques, which allows the anthropometrical walking of a 1.45 m human.

*Corresponding author. Email: marbulu@ing.uc3m.es

Nowadays, the development of humanoid robots has become a very active area. However, it is still limited by the very high cost of maintenance and development.

The main parts of the hardware of the humanoid robot are the ‘custom-built’ components. The software also does not have any standardisation or common rules for the humanoid robot’s programming. It implies the growth of usage of some technologies from the industrial automation field in humanoid robotics because of their low cost and reliability.

The control system of the Rh-1 robot was designed using the conventional electronic components of the automation industry in order to reduce the development time and cost and to have a flexible and easily upgradeable hardware system.

While generating walking patterns we can compute joint angular speed, acceleration and torque ranges (Stramigioli et al. 2002; Arbulú et al. 2005a). There are two methods for designing gait patterns: the distributed mass model and the concentrated mass model. In our case, the concentrated mass model is used because the humanoid dynamics is simplified significantly (Gienger et al. 2001; Kajita et al. 2003a). In order to obtain a natural and stable gait, the 3D-LIPM method is used, where the pendulum mass motion is constrained to move along an arbitrarily defined plane. This analysis leads us to a simple linear dynamics: the three-dimensional Linear Inverted Pendulum Mode (3D-LIPM, Kajita et al. 2003b). Furthermore, the sagittal and frontal motion can be studied in separate planes (Raibert 1986; Arbulú et al. 2006). The 3D-LIPM takes into account that the pendulum ball moves like a free ball in a plane following the inverted pendulum laws in the gravity field, so the ball motion has only three parameters: gravity, the plane position and the ball position. This model is applicable only during the single support phase. Another mass-concentrated model is used, which is the Cart-table model (Kajita et al. 2003a), implemented in order to improve the walking patterns because the Zero Moment Point (ZMP) position can be predicted and a closer relationship with the centre of gravity (COG) is obtained. Smooth patterns by optimising jerk are obtained; this will be seen in successful experiments. In order to apply the obtained trajectory to the humanoid robot Rh-1, the ball or cart motion drives the middle of the hip link. The foot trajectories are computed by single splines, taking into account foot position and orientation and the landing speed of each foot to keep the humanoid from falling down. Several direction patterns will be computed. Some simulation experiments have been done using a 21 DOF VRML robot model.

Then, using the Lie-logic method, the inverse kinematic problem for the entire robot body was solved and the trajectory vectors for each joint $q_i(t)$ were obtained (Paden 1986). These trajectories are used as reference calculated motion patterns and denote feet, arms and the entire body trajectory of the robot.

An industrial robot usually operates in a well-defined environment executing pre-programmed tasks or movement patterns. In the same way, the first approaches to make a humanoid robot walk (Shin et al. 1990) were based on the generation of stable offline patterns according to the ZMP concept (Vukobratovic and Juricic 1969). In contrast to industrial robots, a humanoid robot will interact with a person in a continuously changing workspace. Therefore, the use of only static motion patterns for humanoid robot interaction is insufficient.

The other method is real-time control based on sensor information (Furusho and Sano 1990; Fujimoto et al. 1998). This approach requires a large amount of computing and communication resources and sometimes it is not suitable for a humanoid robot with a high number of joints.

A humanoid robot can walk smoothly if it has previously defined walking patterns and an ability to react adequately to the disturbances caused by imperfections in its mechanical structure and irregular terrain properties. Previous works (Park and Cho 2000; Huang et al. 2000; Yamaguchi et al. 1999) rarely considered and published the detailed architecture for the online modification of dynamic motion patterns.

And finally, a new humanoid platform Rh-1 (see Figure 1) was constructed and successfully tested in a series of walking experiments.



Figure 1. Rh-1 humanoid robot.

This paper presents the control architecture that combines use of the previously offline-calculated motion patterns with online modifications for dynamic humanoid robot walking.

The main contributions of this work are as follows:

- An alternative design of the human-size humanoid robot.
- Validation of the novel approach for walking pattern generation: Local Axis Gait Algorithm (LAG). It permits walking in any direction and on an uneven surface (i.e. ramp, stairs).
- Implementation and validation of the use of screws provide a very geometric description of rigid motion, so the analysis of the mechanism is greatly simplified; furthermore, it is possible to carry out the same mathematical treatment for different robot joints: revolute and prismatic.
- Development of hardware and software control architecture for the humanoid robot Rh-1. This allows us to obtain a more flexible and adaptable system capable of changing its properties according to user needs. Proposed hardware architecture is a novel solution for the area of humanoid robots that complies with modern tendencies in robotics. Software architecture providing the robot with a standard functionality is easily upgraded and can use new one.
- Definition of purpose and validation of kinematic modelling of humanoid robots using screw theory and Paden-Kahan sub-problems having the following advantages:
 - (i) They avoid peculiarities because they offer a global description of rigid body motion; we only need to define two frames (base and tool) and the rotation axis of each DOF, to analyse the kinematics in a closed way.
 - (ii) The Paden-Kahan sub-problems allow for computing the inverse kinematics at position level.
 - (iii) There is a faster computation time of the inverse kinematics with respect to the inverse Jacobean method, Euler angles or D-H parameters, so it contributes to real-time applications.
- Implementation and validation of the inverted pendulum and Cart-table-based walking patterns for any humanoid robot, under the LAG algorithm.
- Development of new efficient algorithms for joint motion control and stabilisation of the humanoid robot gait. These algorithms provide simple solutions allowing for fast and reliable integral control of a robot.

This paper is organised as follows. Section II deals with the human biomechanics study. Section III describes mechanical design and sections IV, V and VI consider hardware, software and communication infrastructures of the humanoid robot Rh-1. Then, Section VII presents walk-

ing pattern generation. Section VIII considers the control architecture implemented in order to control the robot's motion and stability. Simulation and experimental results have been shown in Section IX, and finally, Section X presents conclusions of this work.

II. BIOMECHANICS

A. Outline

The humanoid design starts from its motion requirements, so dimensions, joint range motion, joint velocities, forces and wrench should be studied. After that, the link design can start. This first humanoid robot prototype deals with the study of locomotion, so human locomotion will be analysed. First, human biomechanics anthropometry is studied; next, human walking motion is analysed.

B. Kinematics

The term used for these descriptions of human movement is kinematics. Kinematics is not concerned with the forces, either internal or external, which cause the movement, but rather with the details of the movement itself. In order to keep track of all the kinematic variables, it is important to establish a convention system. Thus, if we wish to analyse movement relative to the ground or the direction of gravity, we must establish a spatial reference system (Figure 2).

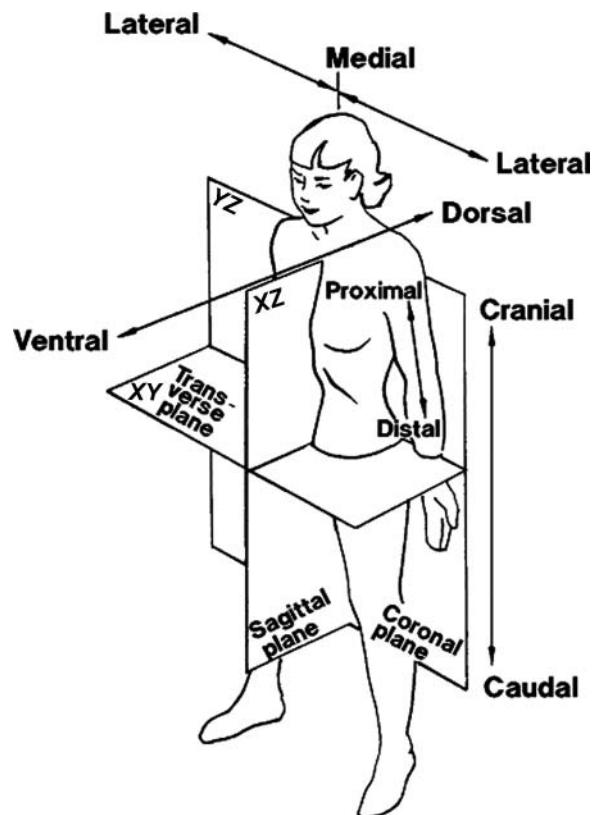


Figure 2. Human motion planes, © NASA.

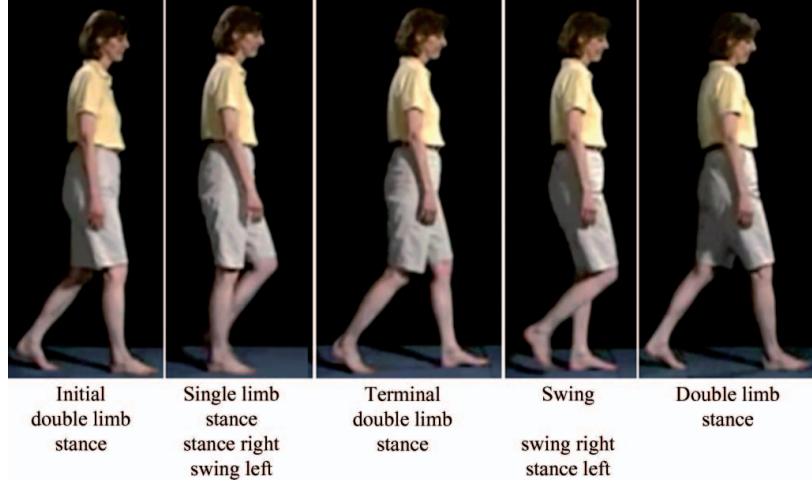


Figure 3. The gait cycle has two phases: about 60% stance phase and 40% swing phase with two periods of double support that occupy a total of 25–30% of the gait cycle.

C. Human locomotion

For dividing the gait cycle in many stages or events, some considerations are taken into account, such as the fact that the gait cycle is the period of time between any two identical events in the walking cycle (Ayyappa 1997). As the gait cycle could be divided into events and the continuity among them must be maintained, any event could be selected as the starting of the gait cycle (that is in the ideal case because the terrain imperfections and human postures make gait cycle not periodic, see Figure 3). So, the starting and finishing events are called the initial contacts respectively. Otherwise, the gait stride is defined as the distance between two initial contacts of one foot.

The stance and swing are the events of the gait cycle. Stance is the event when the foot is in contact with the ground (around 60% of the gait cycle). Swing is the event when the foot is in the air (around 40% of the gait cycle).

D. Anthropomorphic human dimensions, volume and weight distribution

Human dimensions are taken into account as a reference because their proportions allow for stable walking and optimal distribution of forces actuating while a human is walking. Biomechanics give us the relationship between human height and each link (Figure 4, Winter 1990), as well as in the same way as the mass.

E. Human walking trajectories

Human walking motion is studied in order to analyse the right motion of each link and joint during the step. The swing leg and hip motions must assure stable walking in any direction and speed.

The joint angular evolution during a walk should be measured with the appropriate devices, or by introducing the swing leg and hip trajectories as inputs of a kinematic model. For the humanoid robot, joint angular evolution is the input for walking. The human swing foot normally falls to the ground when walking, while for a humanoid robot this must be avoided in order to protect the robot structure and force sensors of the soles. Thus, an adequate walking pattern should be generated for the COG and the swing foot. Normally, the human COG follows the laws of the inverted pendulum in the field of gravity during the walking motion, which is a hyperbolic orbit. It is suitable for making a smooth walking motion at the jerk level. However, the humanoid robot's swing foot motion should be faster than the human one in order not to fall while walking.

Figures 5 and 6 show the leg motion and the hip, knee and feet trajectories (including the ankle, toe and heel). The hip trajectory is quite similar to the COG trajectory. In the sagittal view, that trajectory climbs and descends cyclically. The falling motion increases the sole reaction force, so in the humanoid robot it is better to have a motion on a horizontal plane; furthermore, the trajectory shape looks like the inverted pendulum motion (top view, Figure 6), so we could approximate the humanoid robot in this way.

III. Mechanical design

A. Starting point

The design began with a very thorough study of the ‘state-of-the-art’ humanoid robots made before now. In this case, special attention was paid not only to the form, number of degrees of freedom (DOF), their distribution, structural

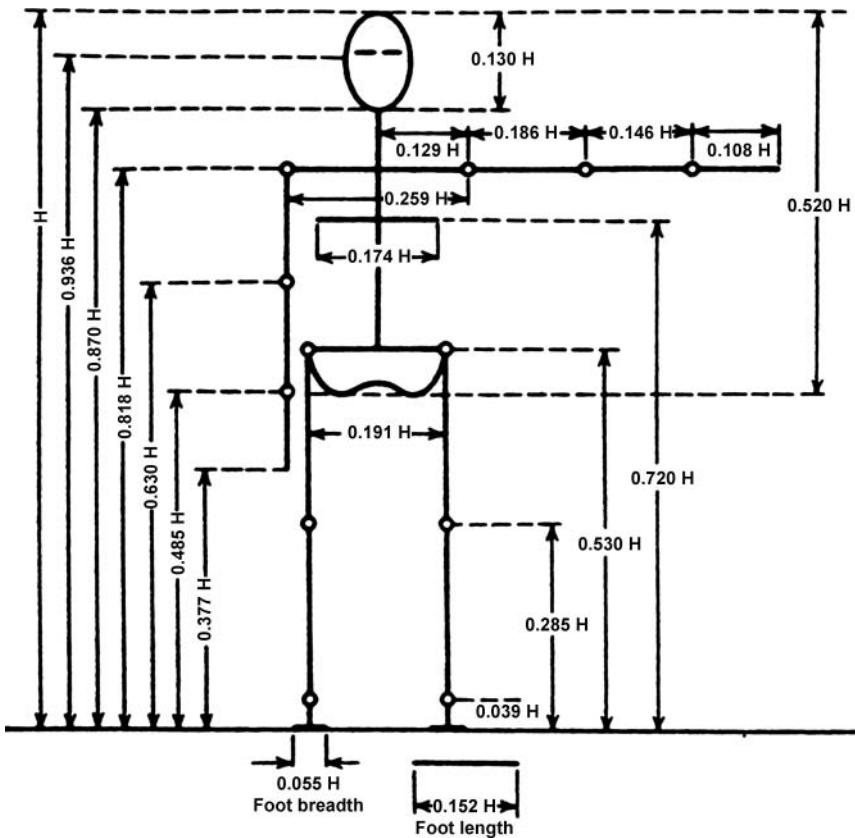


Figure 4. Anthropomorphic human dimensions, © Winter, DA, 1990.

disposition of actuators and characteristics of each one but also fundamentally what goal each one of them served, for example a robot designed for elevating loads has a very different engineering analysis than a robot that only serves people (Hirai et al. 1998).

As a result, it was stated that the primary objective of the prototype Rh was the accomplishment of a humanoid robot with a height of up to around 1.20 m, which was able to walk in any direction with a variation of heights, able to manipulate light objects of up to 750 g with the end of



Figure 5. Human leg motion, sagittal view.

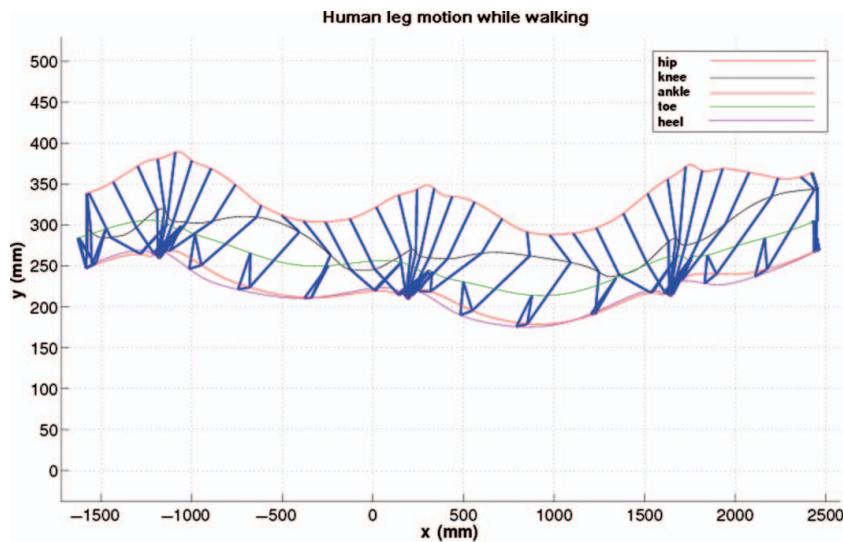


Figure 6. Human leg motion, top view.

its arm and recognise by means of sensors located in its head the place towards which it has to go, as well as voice instructions.

Thus, we arrived at the basic configuration of the robot, starting off with an anthropomorphic design, equipped with two legs, each one with 6 rotational degrees of freedom (DOF). In addition, three of these generate movements in the sagittal plane (lower hip, knee and ankle), two in the frontal plane (ankle and middle hip) and one in the cross-sectional plane (upper hip). Above the hip, we have the trunk where the hardware is located, and two arms. The humanoid has 9 DOF in the upper part of the robot, the trunk has 1 DOF in the neck and in the arms there are 4 DOF that are distributed in the following way: 2 DOF in the shoulder, 1 DOF in the elbow and the last DOF in the wrist, as can be observed in Figure 7.

This configuration allows the biped to walk effectively.

The design parameters require careful consideration and planning to secure walking stability.

In Figure 8(a) we can see the prototype created in a CAD/CAM program that is almost finalised, previous to its entrance to the factory, after adjusting to the maximum all the constructive and engineering parameters that we considered. Figure 8(b) shows the first mounted prototype of the Rh-1.

Finally, Rh-1 is the humanoid robot built as a research platform to perform stable biped walking. Once the first stage was successfully accomplished, the work centred on interaction with humans and the typical environment (Cabas et al. 2004).

B. Robot's design: general considerations

The robot's design implies the creation of specifications that allow it to walk in a dynamic way.

Individual design parameters, such as the length and the mechanics or the distribution of masses, have a greater effect on human-like walking.

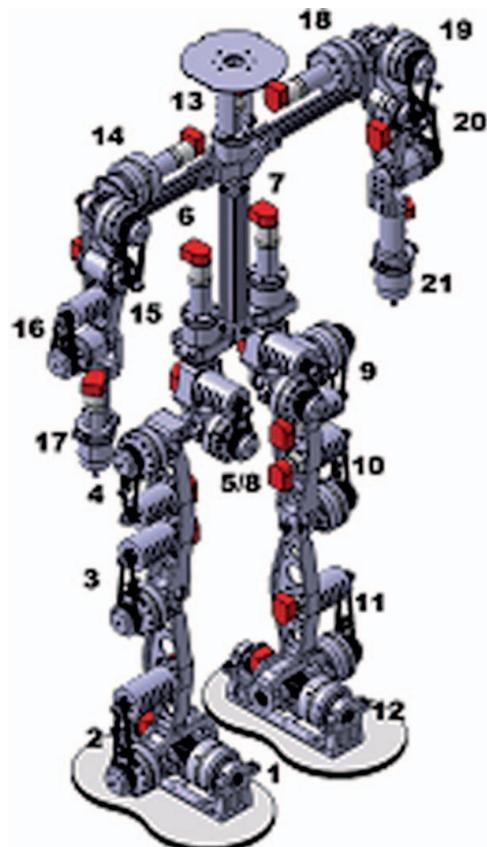


Figure 7. Mechanical configuration of Rh-1 robot.

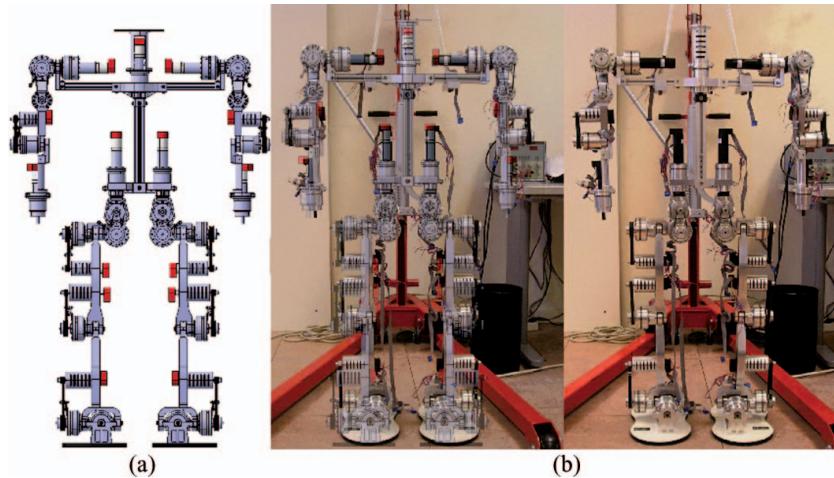


Figure 8. Development of Rh-1: (a) Development in CAD/CAM, and (b) final prototype.

Therefore, a correct selection of proportions of the robot such as the distance between DOF and the distribution of masses across the body are essential for the success of a good design (Cabas et al. 2006a).

By such reasoning, a very important objective to consider at the time of robot design was that it should have the suitable proportions that would allow it to balance itself with relative facility.

With respect to this subject, it should be noted that in the lower part of the robot it was chosen to have equal distances between the ankle (DOF N°2) and the knee (DOF N°3), as well as between the knee (DOF N°3) and the lower hip (DOF N°4) in the sagittal plane (Figure 9).

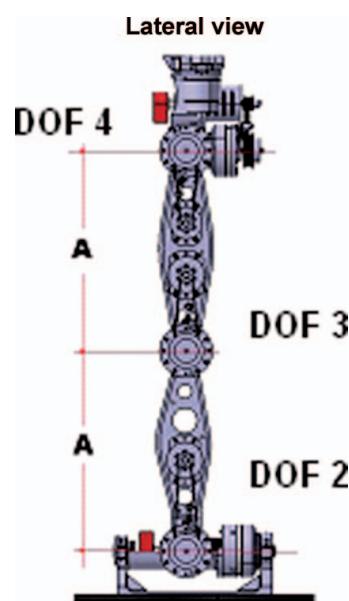


Figure 9. Lower part of mechanical structure.

The importance of the distribution of masses within the robot is a decisive factor at the time of controlling stability, as is the inertia. The joints of the robot move following a certain gait; this means that the COG position has significant influence on the stability of the robot, which demonstrates the importance of the correct distribution of masses to be able to make walking dynamically coherent and with the lowest power consumption possible. This concept is fundamental when designing a robot.

When developing a correct form of dynamic walking without neglecting stability, it is very important to correctly project the joints where the crossing of axes takes place; these are called 'joints of rectangular axes' (JRA).

These joints mechanically represent several DOF in a very small space. They implement the feature that these DOF belong to different planes and their axes will be crossed at a point. To establish a hierarchical structure it might be said that there are JRAs of second order when 2 DOF simultaneously cross at a point and of third order when 3 DOF do so.

The Rh-1 has several joints of rectangular axes. According to Figure 10, the hip joint consists of three rectangular axes (mechanically 3 DOF or JRA of third order), whereas the ankle joint consists of two rectangular axes (mechanically 2 DOF or JRA of second order); also in the shoulder we have a rectangular joint of two axes (mechanically 2 DOF or JRA of 2nd order).

As can be seen in Figure 10, a mechanism of compact design was obtained that allows different implied rectangular axes to not only be able to make the movement without problems but also with the necessary rigidity. At the present time, the possibility of including an additional DOF located in the neck is being considered which would allow for a nodding movement of the head and another one for allowing the displacement of the batteries in the frontal plane.

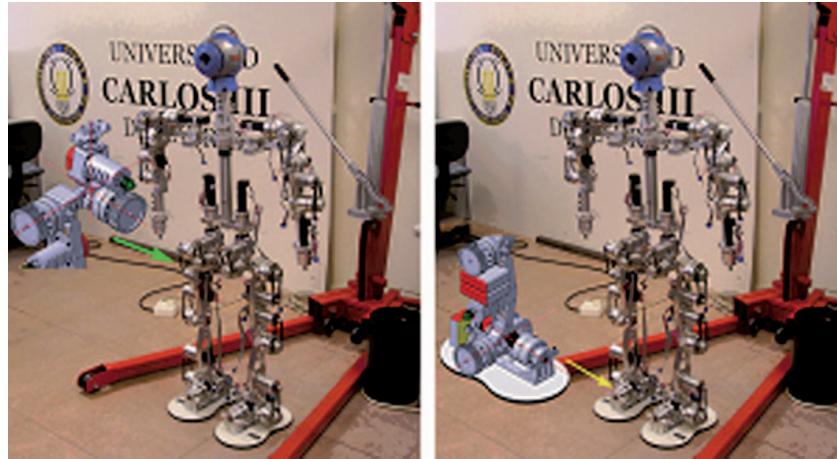


Figure 10. Rh-1's joints of rectangular axes.

Within the chosen scheme, note that the arrangement of the shoulder joints of the shoulder and ankle, in which the axes of the frontal and sagittal planes have been crossed to simplify the kinematics, optimise the movement and also allow for the most anthropomorphic gait possible, as well as maintain the pelvis configuration in Cantilever or beam in projection, providing the robot with wide motion range. Table 1 shows the final mechanical specifications of the Rh-1 humanoid robot.

C. Mechanical study

At the beginning of the project it was necessary to consider a large number of constructive mechanical aspects of the robot, for example the limitations of movement (see Table 1), loads to which the structure will be exposed, torques that the motors must be able to support in each joint etc.

To define the movement of an artificial mechanism with two feet, it is always necessary to make some certain simplifications; some forced ones (e.g. where a motor acts like

several muscles, and geared reduction as in Figure 11) and others for practical reasons.

We can say that although for the human beings the action of walking is something mechanical and practically intuitive, the detailed study of the human gait and why it is done that way, is something complex and difficult to describe. For our part, by means of a complex kinematic analysis, we studied the trajectories that our humanoid would have to follow, where a number of considerations were established

Table 1. Rh-1 specifications.

	Head		—
	Arms	4 DOF/arm (x^2)	8
D		Shoulder 2 ($x2$)	(R, P)
		Elbow 1 ($x2$)	(P)
O	Hands	Wrist 1 ($x2$)	(Y)
		1 ($x2$)	2
F	Torso	1	(Y)
	Legs	6 DOF/Leg ($x2$)	12
		Hip 3 ($x2$)	(R, P, Y)
		Knee 1 ($x2$)	(P)
		Ankle 2 ($x2$)	(R, P)
	<i>Total</i>		21 without hands
D	Height		1450 Mm
I	Wide		300 Mm
M	Deep		230 mm
E	Arm length		200 mm
N	Forearm length		200 mm
S	Hand length		100 mm
I	Femur length		276 mm
O	Tibia length		276 mm
N	Ankle height		60 mm
S	Foot		330 × 200 mm
W	Head		1 Kg
E	Arms	4,5 Kg/arm ($x2$)	9 Kg
I	Bodies		20 Kg
G	Legs	7,6 Kg/leg ($x2$)	15 Kg
H	<i>Total</i>		45 Kg

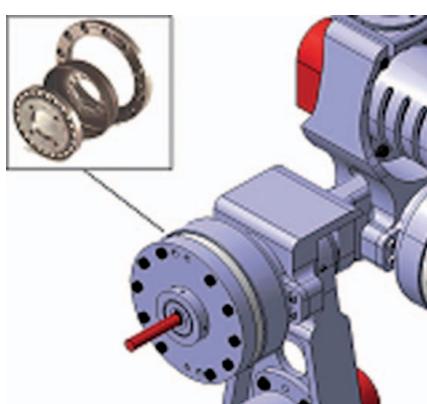


Figure 11. Mounted gear set.

in order to be able to get the optimal trajectory, without which the robot would lose its balance and fall down. For the same reason, the concept of the ZMP was taken into account at the time of designing the movement of the robot.

This study, obviously, was approximated, but the obtained results serve to give an orientative idea of the torques that are going to take place in the critical joints such as the ankles and the knees of the biped. Therefore, it becomes necessary to make a dynamic study with the objective of calculating the torsion efforts that will affect the joints, so that when the motors that are selected start to move the robot, they are the most optimal possible, neither over-sized nor prone to failure due to efforts that they are incapable to withstand (de la Torre et al. 2004).

D. Selection of actuators

Once all the necessary data for each joint has been collected, the selection of the motion system begins. Starting with the motors, at the time of choosing there are two factors that outweigh the rest: first, the minimal possible weight, and second, maximal ratio nominal final torque/weight. The others are important, but absolutely different. Thus, within the great variety of motors presented in the market, we chose to equip the robot with the 'DC-Micromotors Graphite Commutation type' ones. For practical reasons, the entire robot, with its 21 DOF is equipped with only three different types of these motors. It is obvious that these motors do not provide the necessary torque in each joint's axis; for that reason it was decided to equip each one of them with a gear set (see Figure 6). This set is composed of a gear head and in some cases a belt transmission where the right choice had to be made, given the space available and the joint's torque. After an exhaustive analysis considering several parameters and gear-producing companies, it was decided to use the Harmonic Drive AG gear family for all joints. In order to complete this analysis we can say that in spite of a presumably more competitive price, the use of other technologies in the design, such as conventional planetary reducers or cycle technology absolutely does not compensate for their lower quality.

It should also be mentioned that in all the joints that consist of belt transmission, it is noteworthy that one was made via parallel axes. Given that the humanoid robot is a machine that has a mechanical movement with static loads and dynamic and repetitive cyclical changes, it turns out that deciding on another type of transmission (e.g. by pinion and a crown at 90° or a rack) would be very troublesome and therefore not very effective. The dynamic analysis of the upper structure of the humanoid robot does not require estimative methods for the torques calculation since it does not present a closed chain. The only particularity is the one located in the neck joint, which simultaneously generates the movement to turn both arms around the body.

Finally, the structural calculation is carried out separately for each leg, obtaining the torques that the motors of each joint are going to support. With the results obtained, the effective torque that the motors should give during the evolution of the gait for each method was calculated. The choice of the motors needed for the biped was made always taking into account as a critical value the maximum effective torque between both the methods, in order to assure its resistance.

Since a humanoid robot will interact with humans, it must be safe against possible accidents. It is already possible to use a class of actuators with variable mechanical stiffness (Bicchi et al. 2004). The possibility of varying stiffness during motion is a useful way to guarantee low levels of injury risk during execution of fast humanoid robot joint trajectories. Thus, a future aspect of mechanical design would be concerning safety in human–humanoid robot interaction.

E. Structural analysis

It is of great importance to know whether or not the structure will be able to support the weight of the large amount of mechanical, electrical and electronic components that the robot will have onboard. During structural analysis, a measurement of all the structures was made with its later verification by means of the method of finite elements (FEM). Before we enter into detail, it is helpful to mention that the entire structure of the robot was made from aeronautical aluminium 7075.

Then, during this verification, the analysis of the key pieces, which are, in our estimation, the most important within the structure of the robot, will be presented. In the six analysed pieces, loads and moments proportional to the weight of the structure that they support have been considered, adding a safety factor of 2, applying them to the points where the maximum efforts occur (Cabas et al. 2006b).

In the analysis of the ankle (a similar methodology was followed for all pieces), loads equivalent to the whole weight of the robot were applied to each contact surface with the support of the axis, simulating what in fact was to hold all the weights of the structure during the simple support phase. The load has been considered to be 45 kg, which multiplied by a safety factor 2 gives 90 kg, i.e. about 900 Newtons, which have been distributed on two halves on each side of the ankle, where the lodgings of the axes are located. Case A (Figure 12), with 'fillets', is with radius of curvature (therefore eliminating the effect of stress concentration), and case B is without radius of curvature. Case C and case D correspond to flexo-torsion efforts.

In the analysis of the tibia (Figure 13), loads in contact with the thigh were applied, with angles of 10° in case A and 30° in case B as these are the initial and the final working angles for each case. In case C and case D, because these

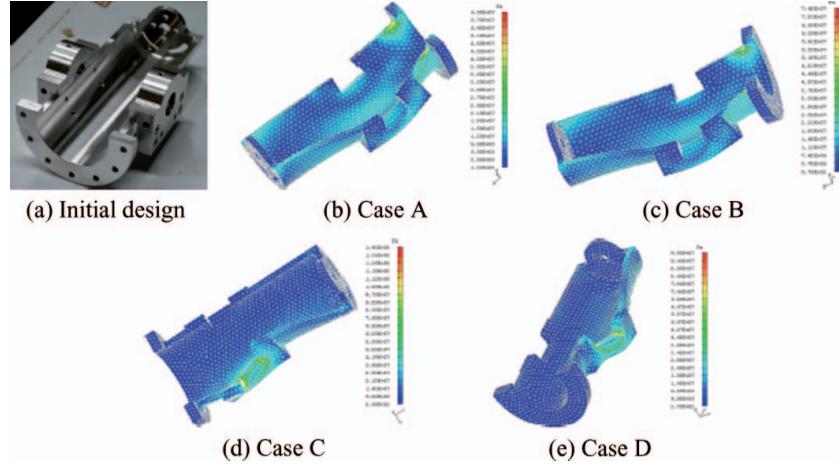


Figure 12. Ankle structural analysis.

represent the flexo-torsion efforts as mentioned above, the weight of the entire robot is also considered, multiplied by a safety factor.

For the analysis of the hip (Figure 14), we divided it into the upper and lower hip. In the analysis of the lower

hip, one of the most critical and difficult pieces to conceive, we applied a load simulating a lateral effort in case A, and in case B another one that represented the weight of the structure (considering a structure of weight equal to that of the whole robot). In the analysis of the upper hip,

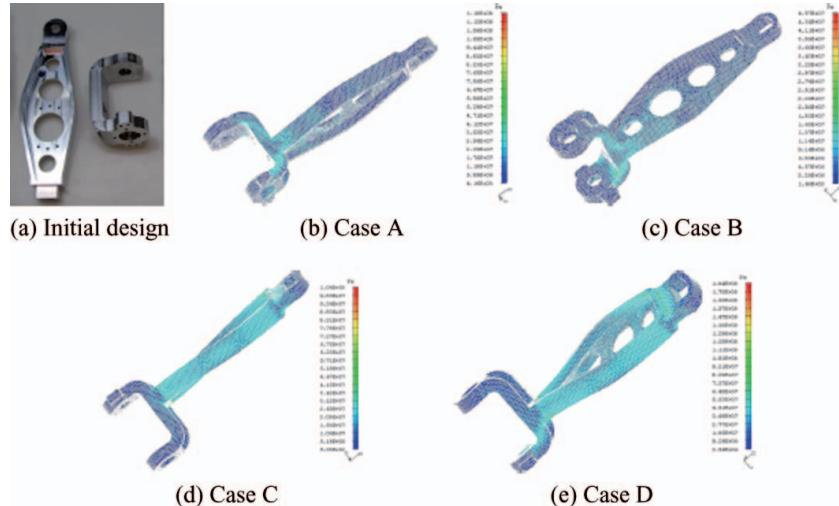


Figure 13. Tibia structural analysis.

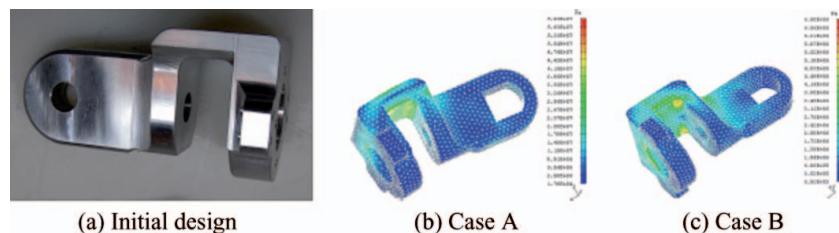


Figure 14. Hip structural analysis.

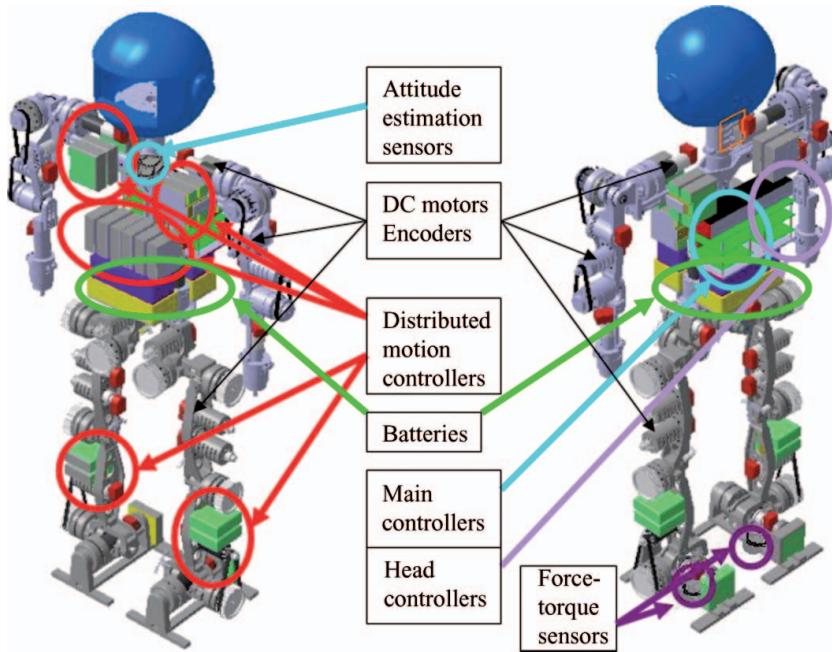


Figure 15. Hardware distribution inside the humanoid robot.

compression loads were applied in case A, and traction loads in case B. Case C represents the effect of loads applied to the union with the lower hip.

The objective of this analysis was to verify that the structure was sufficiently resistant to support different types of movements that are made by the robot. From the analysis carried out with the chosen structure for our robot, experimental results were obtained that include the final design as well as the humanoid robot structure, the chosen systems of performance and the outer appearance, although this last one has been slightly modified due to manufacturing reasons.

IV. Hardware architecture

The hardware architecture for the humanoid robot has some important restrictions imposed by the limited availability of space. In general, the basic requirements for hardware architecture of a humanoid robot are scalability, modularity and standardised interfaces (Regenstein and Dillmann 2003). In the case of the Rh-1 robot with 21 DOF, which assumes the use of 21 DC motors in synchronised high-performance multi-axis application, it is the first necessity to choose an appropriate control approach. The trend of modern control automation is towards distributed control. It is driven by one basic concept, i.e. by reducing wiring, costs can be lowered and reliability increased. Therefore, the electrical design of the Rh-1 robot is based on a distributed motion control philosophy where each control node is an independent agent

in the network. Figure 15 shows the physical distribution of the hardware inside the humanoid robot.

The architecture presented is provided with a large level of scalability and modularity by dividing the control task into Control, Device and Sensory levels (Figure 16) (Kaynov et al. 2005).

The Control level is divided into three layers represented as a controller centred on its own tasks such as external communications, motion controller's network supervision or general control.

In the Device level each servo drive not only closes the servo loop but also calculates and performs trajectory online, synchronises with other devices and can execute different movement programs located in its memory. These forms of devices are located near the motors, thus benefiting from less wiring, which is one of the requirements for energy efficiency; they are lightweight and require less effort in cabling. Advanced and commercially available motion controllers were implemented in order to reduce development time and cost. Continuous evolution and improvements in electronics and computing have already made it possible to reduce industrial controllers' size to use them in the humanoid development project. Furthermore, it has the advantage of applying well-supported and widely used devices from the industrial control field, and brings commonly used and well-supported standards into the humanoid robot development area.

On the Control level, the main controller is a commercial PC/104+ single board computer because of its small size and low energy consumption. It was used instead of a

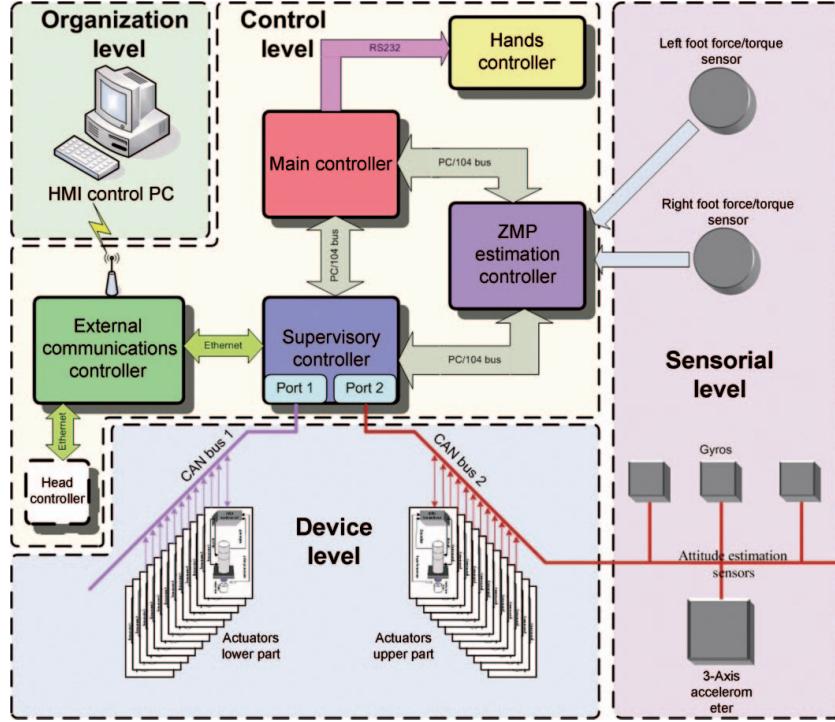


Figure 16. Hardware architecture.

DSP controller because it has a different peripheral interface to the Ethernet and RS-232, and an easy programming environment. In addition, there is a great variety of additional extension modules for the PC/104+ bus like CAN-bus, digital and analogue input-output and PCMCIA cards. Selection criteria were fast CPU speed, low consumption and availability of expansion interfaces. The main controller provides general synchronisation, updates sensory data, calculates the trajectory and sends it to the servo controllers of each joint. It also supervises data transmission for extension boards such as Supervisory Controller and ZMP Estimation Controller via PC/104+ bus.

The communication supervisory controller uses a network bus to reliably connect distributed intelligent motion controllers with the main controller.

The motion control domain is rather broad. As a consequence, communication standards to integrate motion control systems have proliferated. The available communication standards cover a wide range of capability and cost ranging from high-speed networked IO subsystems standards to distributed communications standards for integrating all machines on the shop floor into the wider company. The most appropriate solution to be implemented in the humanoid robot motion control system design seems to be the use of CAN-based standards. The CAN bus communication is used for the Sensory level and the CANOpen protocol on top of the CAN bus is used for the Device level of communication.

Thus, the control system adopted in the Rh-1 robot is a distributed architecture based on the CAN bus. The CAN bus has also been chosen because of various characteristics, such as bandwidth up to 1 Mbit/s that is of sufficient speed to control the axes of a humanoid robot, a large number of nodes (Rh-1 has 21 controllable DOF), differential data transmission, which is important for reducing the electromagnetic interference (EMI) effects caused by electric motors and, finally, the possibility for other devices such as sensors to reside in the same control network.

At the Device level, the controller's network of the Rh-1 is divided into two independent CAN buses in order to reduce the load of communication infrastructure. The lower part of bus controls 12 nodes of two legs and the upper part of bus controls 10 nodes of two arms and the trunk. To unify the data exchange inside the robot, the attitude estimation sensory system is also connected to the upper part CAN bus. In this way, the communication speed of CAN bus used in Rh-1 is 1MBit/s. The synchronisation of both parts is realised by the supervisory controller at the Control level of automation.

The external communications module provides the Ethernet communication on the upper (Control) level of the automation with head electronics, which comprise an independent vision and sound-processing system. It also provides wireless communications with the Remote Client, which sends operating commands for the humanoid robot. The proposed architecture complies with the industrial

automation standards for the design of the motion control system.

V. Software architecture

As mentioned above, a humanoid robot can be considered as a factory where the shop floor consists of a series of cells (intelligent motion controllers and sensors) managed by controllers (the main controller, communication supervisory controller etc.) In general, there are two basic control tasks for the control system of a humanoid robot. The first goal is to control all automation and supervise data transmission and the second goal is to control and monitor the entire floor in order to detect failures as early as possible, and to report on performance indicators. In this context, the humanoid robot Rh-1 is provided with a software system allowing the implementation of the industry automation concepts (Kaynov and Balaguer 2007). The software architecture is based on the server-client model (Figure 17).

For security reasons, the control server accepts the connections of other clients, such as the head client, responsible for the human–robot interaction, only if the master client allows it. If the connection is accepted, the master client only supervises the humanoid robot state and data transmission between the robot and other client, but in the case of any conflict it always has top priority.

According to the server-client model, the humanoid robot is controlled by the passive server, which waits for requests, and upon their receipt, processes them and then serves replies for the client. On the other hand, the server controls all control agents that reside in the CAN bus network. In that case, the control server is no longer a slave; it is a network master for control agents that performs their operations (motion control or sensing) and replies for the server.

As a programmable logic controller (PLC) in the automation industry, the control server is designed and programmed as finite state automata. Figure 18 shows the state

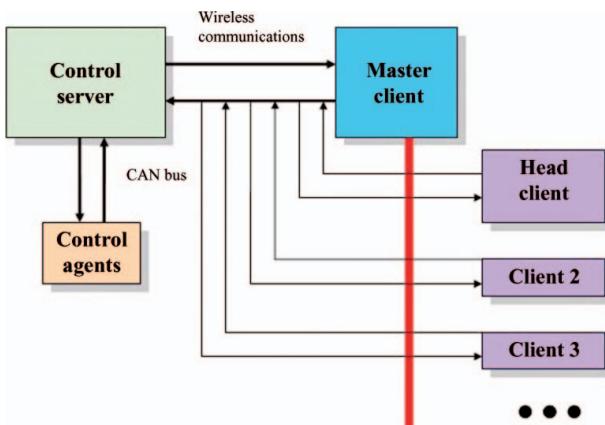


Figure 17. System architecture.

Table 2. State transition events.

Event	Event description
E1	The client is connected
E2	An order has arrived
E3	A command has arrived
E4	A command is sent to the control agent
E5	Agent's reply has arrived
E6	An answer is sent to the client
E7	The user program has successfully terminated or an error event has occurred
E8	Connection with the client is lost
E9	The robot is staying in the secure position
E10	All processes are terminated

diagram and Table 2 provides the state transition events of the humanoid robot control server functioning.

Two basic types of incoming data are processed. A command is simple data, which can be executed by one control agent. The order is a complex command that needs the simultaneous action of many control agents and sensors possessed by the humanoid robot. After the connection of the master client, the humanoid robot stays in the client-handling state waiting for an order or a simple command. The arrival of an order launches the user program. The user program is executed in the control area, the core of the humanoid robot control server software. It performs the data transmission between all control agents, sensory system and the server. It performs trajectory execution at the synchronised multi-axis walking applications, controls the posture and ZMP errors at the dynamic walking mode and reads the sensors' state etc. The control area consists of different modules that provide the execution of motion control for stable biped locomotion of the humanoid robot. All tasks can be grouped by their time requirements.

The developed software provides a set of the C-based functions to work with the robot and generate the user's motions and control procedures that are not only for walking but also for implementing different human–robot cooperation tasks. The code below shows the simple user program. The example in Figure 20 shows how the simple humanoid robot motion can be programmed. At the beginning, the synchronisation procedure for every joint is performed, and then the motion is started. The robot will change the gait (walking mode) according to user request.

In the proposed software architecture, the control server is capable of accepting a large amount of clients' connections at the same time. It is evident that the master client, as the basic HMI of the humanoid robot, should provide and supervise the execution of the upper-level control tasks related with global motion planning, collision avoidance and human–robot interaction. In general, these tasks are common for all mobile and walking robots and the design of these kinds of software systems is not considered in this paper. On the other hand, there are some bottom-level

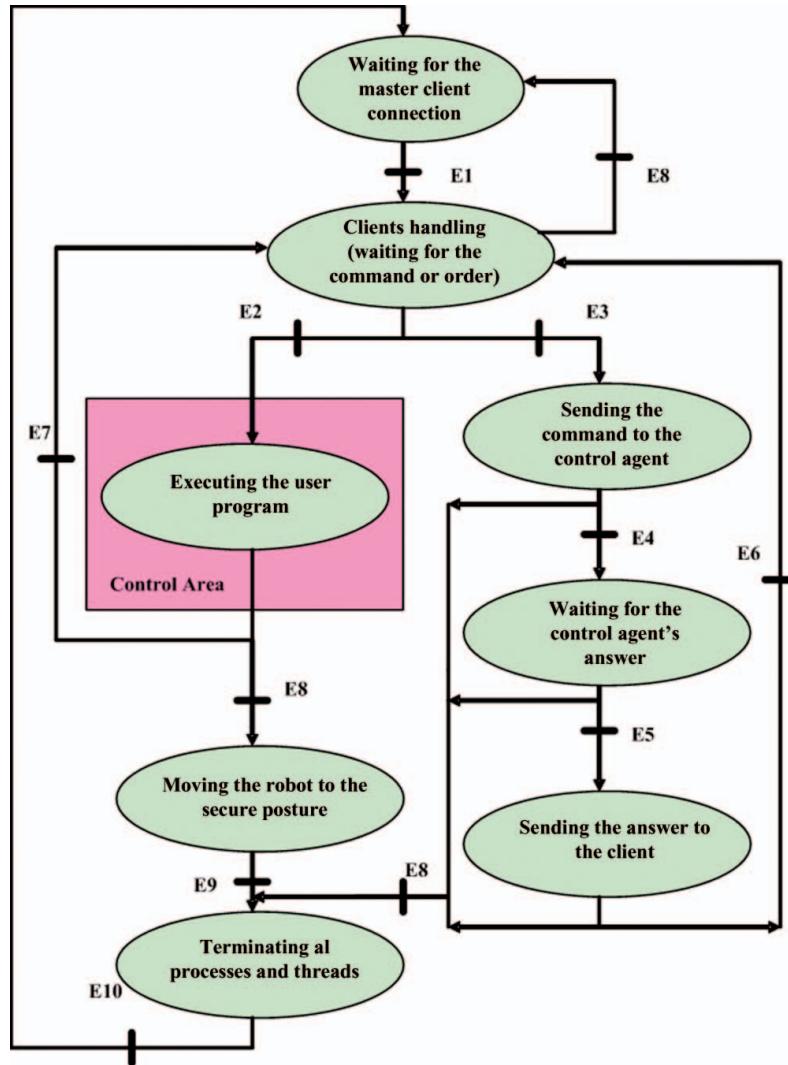


Figure 18. Server-functioning state diagram.

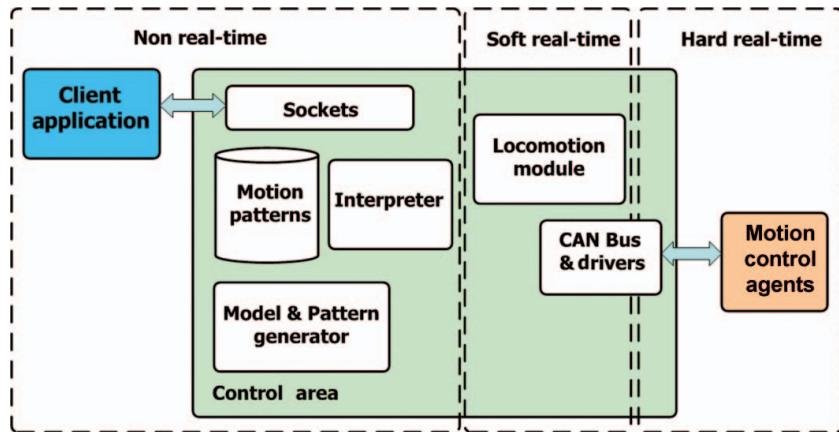


Figure 19. Control area modules.

```
// Motion program example
#include 'control.h'
#include 'sensors.h'
#include 'dynamic_gate.h'

main (void)
{
    EnableRobot ();
    ReadJointsPosition ();
    Sincronize ();
    if (Start ())
        do
    {
        PostureControl ();
        DynamicGate (Gate 1);
        if (ChangeGate)

        DynamicGate (Gate2);
    }
    while (!Stop())
    DisableRobot ();
}
```

Figure 20. Motion program example.

tasks that should be supervised such as sensory data acquisition, joint synchronisation and walking stability control. In order to not overload the master client, which is more oriented to automation supervising, these control tasks are processed with another client application. To provide the robot Rh-1 with bottom-level control, a SCADA system for the humanoid robot, called Humanoid Robot Supervisory Control System (HRoSCoS) Client was developed.

The developed software system is multi-tasking and the control server is also responsible for data acquisition and handling (e.g. polling motion controllers, alarm checking, calculations, logging and archiving) on a set of parameters when the HRoSCoS Client is connected. Figure 21 shows the HRoSCoS Client architecture.

The client requires data or changes control set points by sending commands. The arrival of a command launches its

execution procedure (the right branch of the server functioning state diagram in Figure 18). It consists of the interpretation and transmission of the command to a control agent. When the answer is received, it is converted and transmitted to the HRoSCoS Client to be processed and visualised (Figure 22).

The HRoSCoS Client provides the trending of different parameters of the robot, such as the joint velocities, accelerations, currents, body inclinations, forces and torques which appear during humanoid robot walking. Real-time and historical trending is possible, although generally not in the same chart. Alarm handling is based on limit and status checking and is performed in the control server (e.g. current limit or physical limit of the joint) and then the alarm reports are generated into the HRoSCoS Client application. More complicated expressions (using arithmetic or logical expressions) are developed by creating derived parameters on which status or limit checking is then performed. Logging of data is performed only when some value changes. Logged data can be transferred to an archive once the log is full. The logged data is time-stamped and can be filtered when viewed by a user. In addition, it is possible to generate different reports on the humanoid robot state at any time.

The HRoSCoS Client system presents the information graphically to the operating personnel. This means that the operator can observe a representation of the humanoid robot being controlled (Figure 19).

The human-machine interface (HMI) supports multiple screens, which can contain combinations of synoptic diagrams and text. The whole humanoid robot is decomposed in ‘atomic’ parameters (e.g. a battery current, its maximum value, its on/off status etc.) to which a tag-name is associated. The tag-names are used to link graphical objects to devices. Standard window-editing facilities, such as zooming, re-sizing, scrolling etc., are provided: On-line configuration and customisation of the HMI is possible for users with the appropriate privileges. Links are created between display pages to navigate from one view to another.

VI. Communication infrastructure and methods

When building automation applications, communication with the host is often a crucial part of the project. Nodes of the network always function as data servers because their primary role is to report information (status, acquired data, analysed data etc.) to the host at constant rates.

As shown in Figure 17, hardware architecture consists of three basic levels of automation that uses its own communication systems. The upper (Control) level uses a TCP/IP-based communication protocol. Ethernet communication is one of the most common methods for sending data between computers. The TCP/IP protocol provides the technology for data sharing, but only the specific application implements the logic that optimises performance and makes sense of the data exchange process. When data transmission

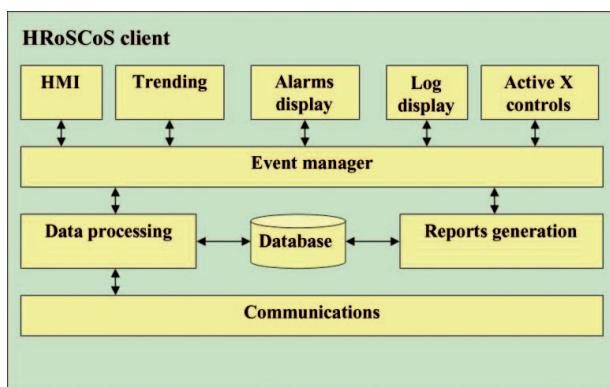


Figure 21. HRoSCoS client architecture.

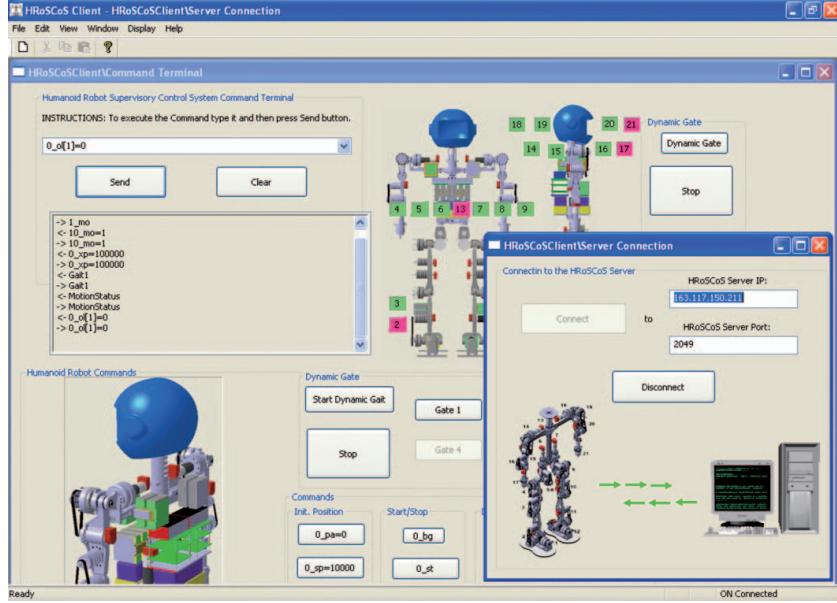


Figure 22. HRoSCoS client views.

begins, the sender should packetise each piece of data with an ID code that the receiver can use to look for the decoding information. In this way, developed communication protocol hides the TCP implementation details and minimises network traffic by sending data packages only when they are needed. When a data variable is transmitted by the sender, it is packetised with additional information so that it can be received and decoded correctly on the receiving side. Before each data variable is transmitted, a packet is created that includes fields for data size, data ID and the data itself. Figure 18 shows the packet format.

The data ID field is populated with the index of the data array element corresponding to the specified variable. Since the receiving side also has a copy of the data array, it can index it to get the properties (name and type) of the incoming data package (see Figure 23). This very effective mechanism is implemented to provide data exchange between the control server and different clients on the Control level of automation of the humanoid robot.

Bottom-level (Sensory and Field) communications are realised using CAN and CanOpen protocols (Figure 24).

These communication protocols provide data transmission of the broadcast type of communication. A sender of information transmits to all devices on the bus. All receiving devices read the message and then decide about its relevance to them. This guarantees data integrity as all de-

vices in the system use the same information. The sensory system of the humanoid robot makes data exchange under lower CAN protocol and the intelligent motion controller uses upper-level CANOpen protocol. The same physical layer of these protocols allows them to reside in the same physical network.

The communication implemented on the bottom level involves the integration of CANOpen (drives and motion control device profile) and the introduction of new functionality, which is not contained within the relevant device profiles for the sensory data processing.

VII. Walking pattern generation

There are many propositions for generating the walking patterns of humanoid robots, some of them a mass distributed-based model (Hirukawa et al. 2007) and other one a mass concentrated-based model (Gienger et al. 2001; Kajita et al. 2003a, b). The first approach describes the motion accurately, but it has a high computation cost, which is not suitable for real-time applications. On the other hand, the second approach saves computation time and performs the walking motion suitably. In this section, two forms of mass concentrated models will be explained and discussed, i.e. the inverted pendulum model and the cart-table model. Both models have been tested on the Rh-1 humanoid robot platform in order to generate stable walking patterns. At first, the 2D inverted pendulum model will be detailed, for introducing pendulum laws; next the 3D version is developed; after that, the Cart-table model will be introduced and its advantages with respect to the

Data Size (8 bits)	Data ID (16 bits)	Data (32 bytes)
-----------------------	----------------------	--------------------

Figure 23. The package format.

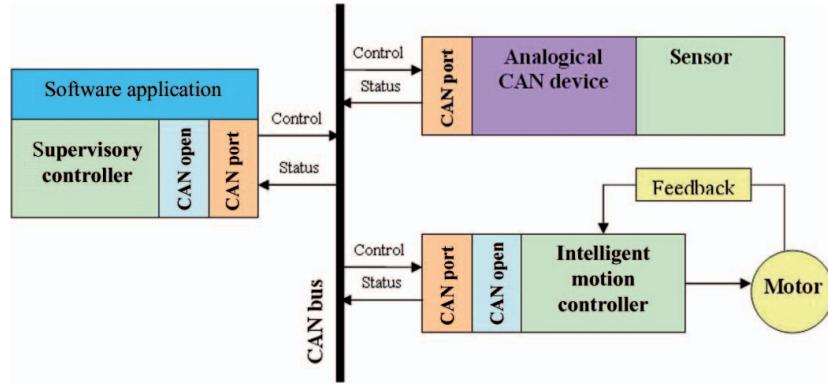


Figure 24. CAN bus-based communication system.

inverted pendulum are explained. Thereafter, the walking pattern strategy is proposed with the ‘LAG’ algorithm (Arbulú and Balaguer 2007, 2009) and finally, in order to compute joint patterns, the inverse kinematics model is proposed by using the screw theory and Lie groups.

A. 2D inverted pendulum model

The gait pattern generation for a humanoid robot could be simplified as studying the motion in the sagittal plane and concentrating all the body mass in the COG. In this way, it is possible to use the 2D inverted pendulum model to obtain stable and smooth walking motion.

The 2D inverted pendulum model is composed of a mass and a telescopic leg without mass (see Figure 25).

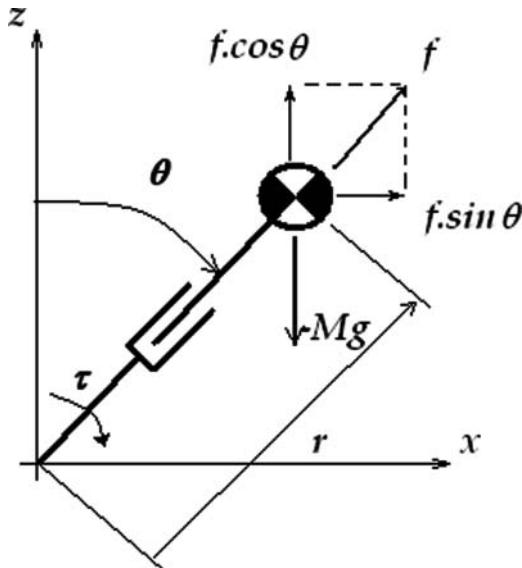


Figure 25. The 2D inverted pendulum model with motion in the $x - z$ plane.

Therefore, the model is described in the next state variables:

r : Radius of position vector (massless and telescopic leg).

θ : Pitch angle.

f : Reaction force on pendulum.

τ : External pitch torque.

From the free body diagram of the pendulum ball, the dynamic equations could be written in the following manner:

$$F_x = f \sin \theta + \frac{\tau}{r} \cos \theta, \quad (1)$$

$$F_z = f \cos \theta - Mg - \frac{\tau}{r} \sin \theta. \quad (2)$$

It is known that $p = (x, z)$, so the dynamic equations of the pendulum ball motion are

$$F_x = M\ddot{x} = f \sin \theta + \frac{\tau}{r} \cos \theta, \quad (3)$$

$$F_z = M\ddot{z} = f \cos \theta - Mg - \frac{\tau}{r} \sin \theta. \quad (4)$$

There are several solutions for the ball pendulum motion from this complex dynamic model. In order to simplify the dynamic problem, some constraints could be taken into account:

1. Motion at constraint height.

$$z = z_c, \quad (5)$$

$$\dot{z} = 0. \quad (6)$$

2. It is possible to consider natural pendulum ball motion, so the input torque turns to zero.

$$\tau = 0. \quad (7)$$

From these constraints the dynamic Equations (3) and (4) reduce the dynamic motion to a linear one:

$$F_x = M\ddot{x} = f \sin \theta, \quad (8)$$

$$F_z = 0 = f \cos \theta - Mg [0, 1]. \quad (9)$$

By combining Equations (8) and (9), the dynamic pendulum ball motion is obtained as

$$\ddot{x} = g \frac{x}{z_c} \frac{n!}{r!(n-r)!}. \quad (10)$$

The natural motion of the pendulum ball depends on the potential gravity field (g), position (x, z) and distance from the pendulum base (z_c). Thus, no linear dynamic motion equations are converted to linear ones; this way a single solution could be found and this kind of trajectory is applicable in real-time applications of walking locomotion.

In order to design walking patterns and determine the spatial geometry of trajectories, the concept of orbital energy is introduced. Orbital energy evaluates the pendulum ball energy at the level of the motion plane. It is composed of the potential and kinetic energy of the pendulum ball. In this way, it is possible to determine whether the pendulum motion is in a state of equilibrium, moves forward or never passes the zero position.

The mathematical expression of orbital energy could be developed by multiplying and integrating Equation (10) by \dot{x} .

$$\dot{x} \left(\ddot{x} - g \frac{x}{z_c} \right) = 0, \quad (11)$$

$$\int \left(\ddot{x} \dot{x} - g \frac{x}{z_c} \dot{x} \right) dt = const, \quad (12)$$

$$\frac{1}{2} \dot{x}^2 - \frac{1}{2} \frac{g}{z_c} x^2 = E. \quad (13)$$

Equation (13) shows that a kind of energy, called *orbital energy* is conserved. The first term represents the kinetic energy per unit mass of the body, while the second one is the virtual energy caused by a force field that generates a force $(g/z_c) \cdot x$ on the unit mass located at x . Furthermore, in Figure 26) $E > 0$, which means that the pendulum mass swings forward; $E = 0$ represents the equilibrium state, i.e. the pendulum mass swinging towards the equilibrium point or the pendulum mass swinging out from the equilibrium point; finally $E < 0$ means that the body never passes point $x = 0$.

At this point, it is possible to generate 2D stable natural walking patterns. This study is the basis for obtaining the solution of 3D walking patterns, suitable for any humanoid robot. Human-like walking motion can be obtained because biomechanical studies demonstrate that COG human motion on the walking cycle could be approached by

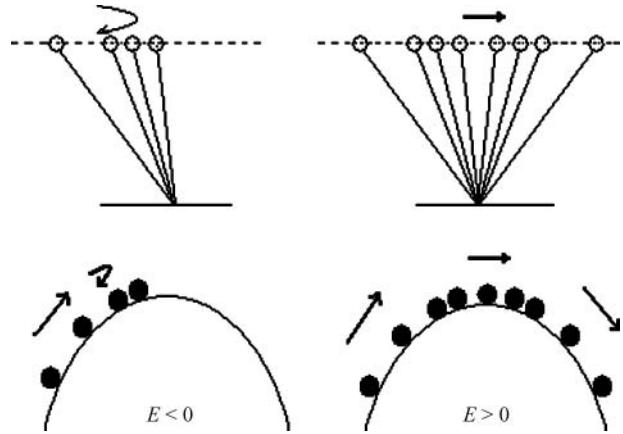


Figure 26. Pendulum ball rolling in potentials.

an inverted pendulum motion. The next section focuses on 3D pendulum motion.

B. Motion laws of 3D-LIPM (inverted pendulum model)

In Figure 27, the 3D inverted pendulum model is shown consisting of a point mass (p) and a massless telescopic leg, where $p = (x, y, z)$ is the position of the mass M , which is uniquely specified by variable $q = (\theta_r, \theta_p, r)$, so that following the right hand rule,

$$x = r \sin(\theta_p), \quad (14)$$

$$y = -r \sin(\theta_r), \quad (15)$$

$$z = r \sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}. \quad (16)$$

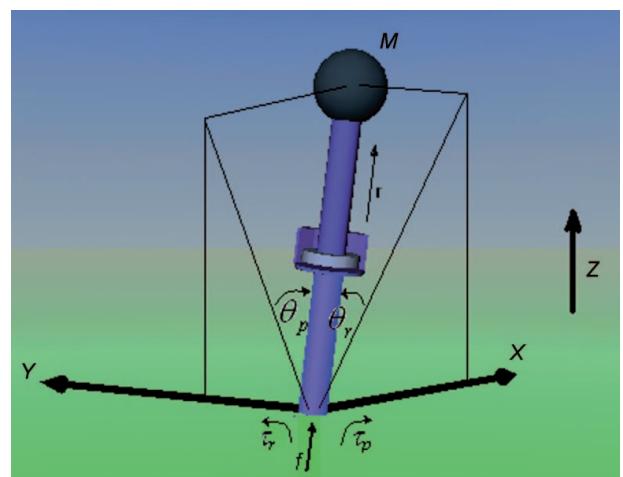


Figure 27. 3D inverted pendulum model.

The motion equation of the inverted pendulum in Cartesian coordinates is as follows:

$$\begin{aligned} & \begin{pmatrix} \tau_r \\ \tau_p \\ f \end{pmatrix} \\ = m & \begin{pmatrix} 0 & -r \cdot \cos(\theta_r) - \frac{r \cdot \cos(\theta_r) \cdot \sin(\theta_r)}{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}} \\ r \cdot \cos(\theta_p) & 0 & -\frac{r \cdot \cos(\theta_p) \cdot \sin(\theta_p)}{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}} \\ \sin(\theta_p) & -\sin(\theta_r) & \sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2} \end{pmatrix} \\ & \times \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} + mg \begin{pmatrix} -\frac{r \cdot \cos(\theta_r) \cdot \sin(\theta_r)}{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}} \\ -\frac{r \cdot \cos(\theta_p) \cdot \sin(\theta_p)}{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}} \\ \sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2} \end{pmatrix}. \quad (17) \end{aligned}$$

So, the dynamics along the x -axis are given by

$$m(z\ddot{x} - x\ddot{z}) = \frac{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}}{\cos(\theta_p)} \tau_p + mgx, \quad (18)$$

and the equation for the dynamics along the y -axis is given by:

$$m(-z\ddot{y} + y\ddot{z}) = \frac{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}}{\cos(\theta_r)} \tau_r - mgy. \quad (19)$$

C. Natural 3D linear inverted pendulum mode (3D-LIPM)

In order to reduce the motion possibilities of the pendulum, we introduce some constraints to limit this motion. One constraint limits the motion in a plane, so that

$$z = k_x x + k_y y + z_c, \quad (20)$$

where z_c is the distance from the xy -plane to the pendulum mass.

Replacing Equation (20) and its second derivative into Equations (18) and (19), we get

$$\begin{aligned} \ddot{x} &= \frac{g}{z_c} x + \frac{k_y}{z_c} (x\ddot{y} - \ddot{x}y) \\ &+ \frac{1}{mz_c} \frac{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}}{\cos(\theta_p)} \tau_p, \quad (21) \end{aligned}$$

$$\begin{aligned} \ddot{y} &= \frac{g}{z_c} y - \frac{k_x}{z_c} (x\ddot{y} - \ddot{x}y) \\ &- \frac{1}{mz_c} \frac{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}}{\cos(\theta_r)} \tau_r. \quad (22) \end{aligned}$$

The above equations allow pendulum motion in any plane and slope, if the motion is constrained to a flat plane ($k_x = 0$ and $k_y = 0$), so that

$$\ddot{y} = \frac{g}{z_c} y - \frac{1}{mz_c} \frac{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}}{\cos(\theta_r)} \tau_r, \quad (23)$$

$$\ddot{x} = \frac{g}{z_c} x + \frac{1}{mz_c} \frac{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}}{\cos(\theta_p)} \tau_p. \quad (24)$$

Note that Equations (23) and (24) are independent equations and no linear dynamics is simplified to a linear one.

The natural 3D-LIPM takes into account the trajectories of the inverted pendulum model without input torques. Hence, Equations (23) and (24) are simplified to

$$\ddot{x} = \frac{g}{z_c} x, \quad (25)$$

$$\ddot{y} = \frac{g}{z_c} y. \quad (26)$$

Solving Equations (25) and (26), a 3D pendulum ball motion is obtained in the gravity field. Figure 28 shows an example of the pendulum motion for a different support

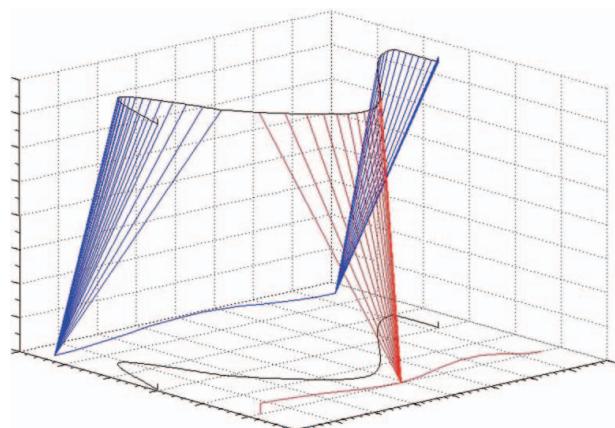


Figure 28. Inverted pendulum motion under natural 3D-LIPM.

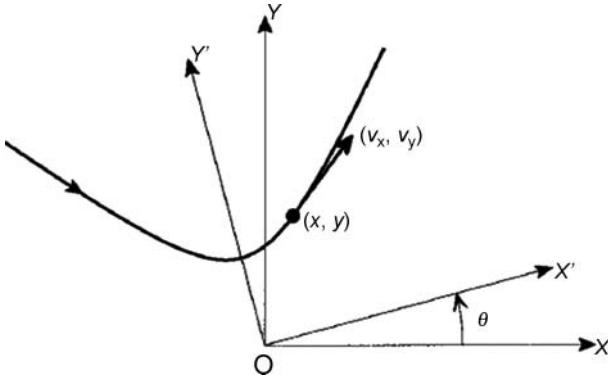


Figure 29. 3D-LIPM projected onto the $x - y$ plane at the local axis.

foot, i.e. blue pendulum motion for the left support foot at its local frame and red pendulum motion for right support foot at its local frame.

D. Geometry of trajectory

The pendulum spatial motion in the gravity field should be studied and analysed in order to predict the stability and suitable 3D local motion. So, describing the local motion (Figure 29) in any rotational axis, it is possible to study the gravitational effects on natural pendulum motion, which is like potential energy acting on a space shuttle.

Orbital energy on each axis being

$$E_x = \frac{1}{2}\dot{x}^2 - \frac{1}{2}\frac{g}{z_c}x^2, \quad (27)$$

$$E_y = \frac{1}{2}\dot{y}^2 - \frac{1}{2}\frac{g}{z_c}y^2, \quad (28)$$

the orbital energy on $X'Y'$ axis is obtained as follows:

$$\begin{aligned} E'_x &= \frac{1}{2}(\dot{x} \cos \theta + \dot{y} \sin \theta)^2 \\ &\quad - \frac{1}{2}\frac{g}{z_c}(x \cos \theta + y \sin \theta)^2, \end{aligned} \quad (29)$$

$$\begin{aligned} E'_y &= \frac{1}{2}(-\dot{x} \sin \theta + \dot{y} \cos \theta)^2 \\ &\quad - \frac{1}{2}\frac{g}{z_c}(-x \sin \theta + y \cos \theta)^2. \end{aligned} \quad (30)$$

We can calculate the axis of symmetry by solving the variation of orbital energy with respect to the rotation angle, and in this way the maximum energy is found; the

mathematical expression is developed as follows:

$$\frac{\partial E'_x}{\partial \theta} = A[(\sin \theta)^2 - (\cos \theta)^2] + B \sin \theta \cos \theta = 0 \quad (31)$$

$$A = \left(\frac{g}{z_c}\right)xy - \dot{x}\dot{y} \quad (32)$$

$$B = \left(\frac{g}{z_c}\right)(x^2 - y^2) - (\dot{x}^2 - \dot{y}^2) \quad (33)$$

Finding the symmetry axis from Equation (31), by trigonometric identities are

$$\frac{A}{B} = -\frac{\sin \theta \cdot \cos \theta}{(\sin \theta)^2 - (\cos \theta)^2} \quad (34)$$

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{2A}{B} \right) \quad (35)$$

It is well known that the y -axis is the axis of symmetry for $\theta = 0$, so that

$$A = \left(\frac{g}{z_c}\right)xy - \dot{x}\dot{y} = 0, \quad (36)$$

$$\left(\frac{g}{z_c}\right)xy = \dot{x}\dot{y}. \quad (37)$$

Equation (37) could be used for computing the 3D-LIPM geometric shape with the orbital energy mathematical expressions from Equations (27) and (28):

$$\left(\frac{g}{z_c}\right)^2 x^2 y^2 = (2E_x + (g/z_c)x^2)(2E_y + (g/z_c)y^2). \quad (38)$$

By simplifying the last equation an interesting expression is found which describes the shape of the pendulum mass trajectory in the gravity field (Equation (39)):

$$\frac{g}{2z_c E_x} x^2 + \frac{g}{2z_c E_y} y^2 = -1. \quad (39)$$

It is possible to deduce that $E_x > 0$, because the x -axis pendulum passes 0 of the local frame and $E_y < 0$, due to the fact that the y -axis pendulum does not pass 0 of the local frame (Figure 26). These facts show us that the pendulum mass trajectory shape is a hyperbolic curve described by Equation (39). Furthermore, the natural pendulum mass motion in three dimensions gives us information about the motion range for several initial conditions, which could be applied to the single support phase of the humanoid body motion.

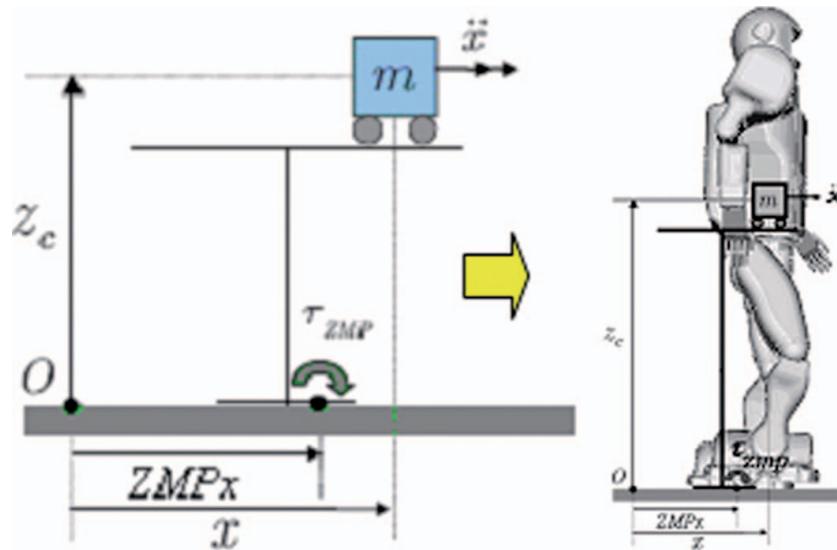


Figure 30. Cart-table model.

E. Temporal equations

With initial conditions (x_i, \dot{x}_i) and (y_i, \dot{y}_i) at time t_i , the mass trajectory is calculated by solving differential Equations (25) and (26):

$$x(t) = x_i \cosh\left(\frac{t - t_i}{T_c}\right) + T_c \dot{x}_i \sinh\left(\frac{t - t_i}{T_c}\right), \quad (40)$$

$$\dot{x}(t) = \frac{x_i}{T_c} \sinh\left(\frac{t - t_i}{T_c}\right) + \dot{x}_i \cosh\left(\frac{t - t_i}{T_c}\right), \quad (41)$$

$$y(t) = y_i \cosh\left(\frac{t - t_i}{T_c}\right) + T_c \dot{y}_i \sinh\left(\frac{t - t_i}{T_c}\right), \quad (42)$$

$$\dot{y}(t) = \frac{y_i}{T_c} \sinh\left(\frac{t - t_i}{T_c}\right) + \dot{y}_i \cosh\left(\frac{t - t_i}{T_c}\right). \quad (43)$$

F. Cart-table model

In order to establish a relationship between the COG and ZMP motion, the Cart-table model is proposed. This model, by controlling Cart acceleration, gives us an interesting relationship between the ZMP and the COG.

By evaluating the torque in the ZMP (Figure 30)

$$\tau_{zmp} = mg(x - ZMP_x) - m\ddot{x}Z_c. \quad (44)$$

As we know, the torque in the ZMP is zero, thus from Equation (44), ZMP_x is by

$$ZMP_x = x - \frac{Z_c}{g} \ddot{x}. \quad (45)$$

Note that Equation (45) is similar to the inverted pendulum (24), with the main difference being that ZMP_x is constrained to 0, while, if we knew it, it would be fixed to any

point on the local axis. In the y direction, a similar equation could be obtained. In order to get the COG motion as an inverse problem from the ZMP one, the solution of Equation (45) should be treated as a servo control problem.

G. Comparing pendulum and Cart-table models

In the inverted pendulum model the input reference is the ZMP and the output is the COG pattern. Note that the ZMP is always at the base of the pendulum (i.e. Figure 31(b)). In

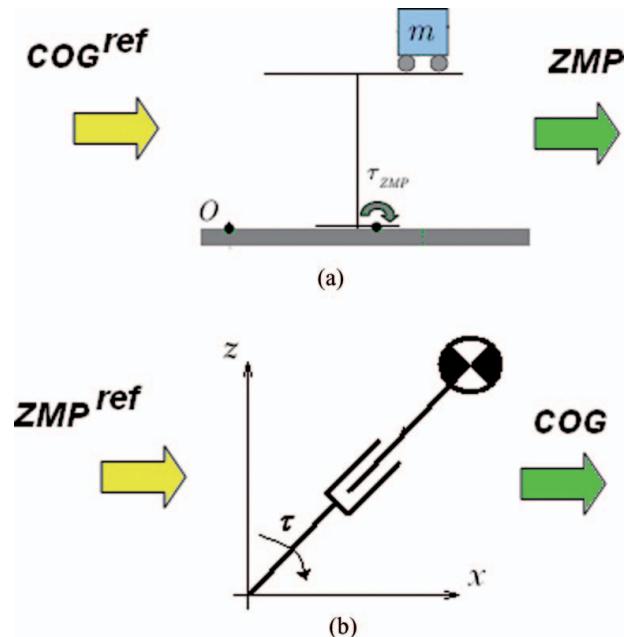


Figure 31. Comparison between (a) Cart-table, and (b) inverted pendulum.

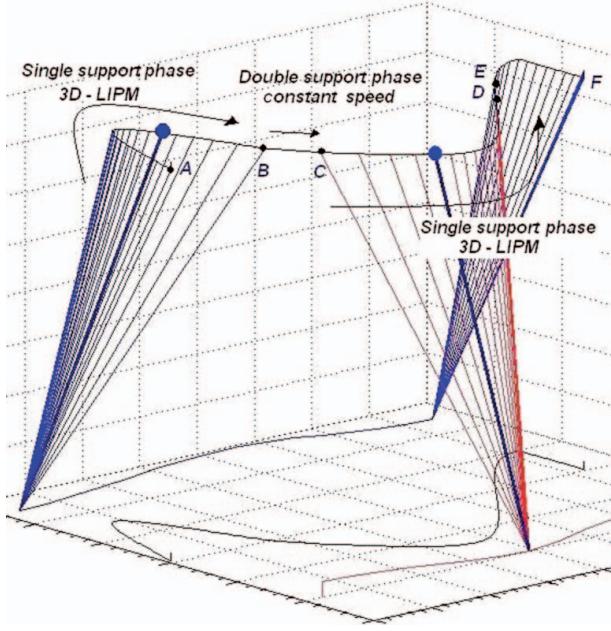


Figure 32. Walking pattern strategy.

the Cart-table model, the ZMP motion is instead generated by the COG as reference (i.e. Figure 31(a)).

Other facts are that there is a discontinuity in the change from a single support phase to a double one, so at high-walking velocity, jerk is an important fact; it could be improved by using high-order splines. Thus, the Cart-table model optimises jerk and continuity is maintained at all times, no matter the change of phase, and in this way a high walking speed is possible.

H. Walking pattern strategy

Figure 32 shows the steps of the walking strategy. In the single support phase the pendulum ball follows 3D-LIPM laws (A to B, C to D and E to F); in the double support phase, the pendulum ball moves at a constant speed (B to C and D to E). This motion drives the COG of the humanoid robot. We could assume that the COG is in the middle of the hip joint. Foot trajectories are computed by single splines taking into account some constraints, such as step length, maximum height of the foot, lateral foot motion, foot orientation and speed in order to avoid falling down and to reduce the impact force on the landing foot (Figure 33).

I. Local axis gait (LAG) algorithm

In order to generalise the walking patterns of any direction and surface, such as stairs or slopes, the ‘Local Axis Gait algorithm’ is proposed, (Arbulú et al. 2007) in order to plan stable local walking motion. The LAG is divided into several stages: computation of the footprints;

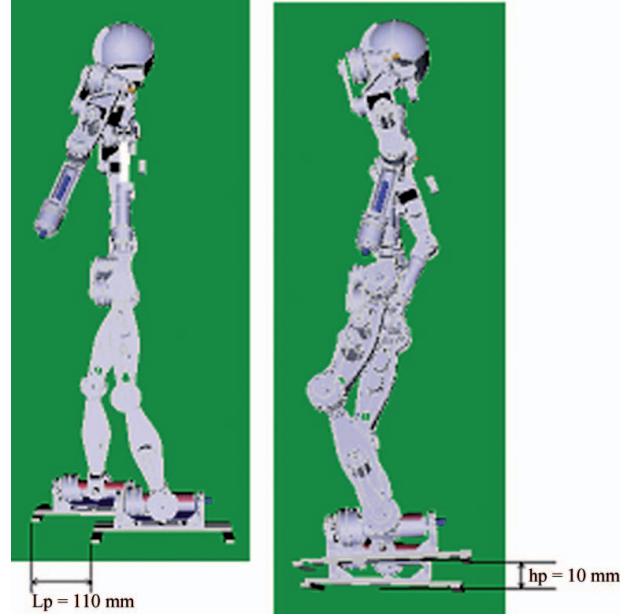


Figure 33. Some foot trajectory constraints: maximum step length and maximum swing foot height.

the decision of the ZMP limits around the footprints; the dynamic humanoid COG motion generation based on the mass-concentrated model; and finally joining the footprints of the swing foot by splines. In this way, it is possible to generate each step online, using the desired footprints as input.

The footprints (Figure 34) for doing an n -th step can be computed as follows:

$$P^n = P^n + R(\theta_z^n)^T \cdot L^n, \quad (46)$$

where

$$\begin{aligned} P^n &= (p_x^n \ p_y^n \ p_z^n)^T, \\ L^n &= (L_x^n \ -(-1)^n L_y^n \ L_z^n)^T, \\ \sum, \sum^n, \sum^{n-1}, \sum^{n+1} &\text{: world and feet frames,} \\ P^n, P^{n-1}, P^{n+1} &\text{: feet position,} \\ L_x^{n+1}, L_y^{n+1}, L_z^{n+1} &\text{: swing foot displacements,} \\ \theta_x^{n+1}, \theta_y^{n+1}, \theta_z^{n+1} &\text{: rotations about world frame.} \end{aligned}$$

The walking patterns developed are introduced into the inverse kinematic algorithm (Arbulú et al. 2005a) to obtain the angular evolution of each joint; these are the reference patterns of the humanoid robot.

J. Inverse kinematic model

In order to compute the robot’s joint motion patterns, some kinematic considerations must be made. Due to the fact that the kinematic control is based on screw theory and Lie logic techniques, it is also necessary to present a basic explanation.

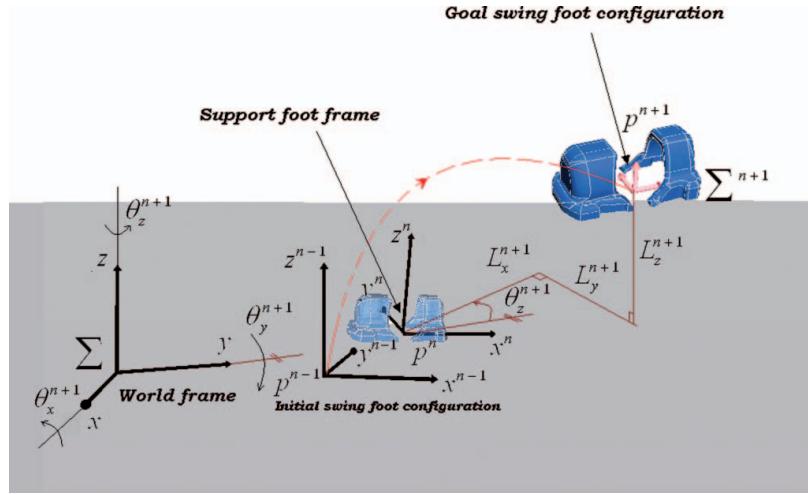


Figure 34. Footprint location.

Lie logic background

Lie groups are very important for mathematical analysis and geometry because they serve to describe the symmetry of analytical structures (Park et al. 1995). A Lie group is an analytically manifold group. A Lie group algebra is a vectorial space over a field that completely captures the structure of the corresponding Lie group. The homogeneous representation of a rigid motion belongs to the special Euclidean Lie group ($SE(3)$) (Abraham and Marsden 1999). The Lie algebra of $SE(3)$, denoted as $se(3)$, can be identified with the matrices called twists ' ξ^\wedge ' (see Equation (47)), where the skew symmetric matrix ' ω^\wedge ' (Equation (48)) is the Lie algebra $so(3)$ of the orthogonal special Lie group ($SO(3)$), which represents all rotations in the 3D space. A twist can be geometrically interpreted using screw theory (Paden 1986), as Charles's theorem proved that any rigid body motion could be produced by a translation along a line followed by a rotation around the same line; this is a screw motion, and the infinitesimal version of a screw motion is a twist.

$$\begin{aligned}\xi^\wedge &= \begin{bmatrix} \varpi^\wedge v \\ 0 \end{bmatrix} \in se(3)/se(3) \\ &= \{(v, \varpi^\wedge) : v \in \mathbb{R}^3, \varpi^\wedge \in so(3)\} \in \mathbb{R}^{4 \times 4},\end{aligned}\quad (47)$$

$$\begin{aligned}\varpi^\wedge &= \begin{bmatrix} 0 & -\varpi_3 & \varpi_2 \\ \varpi_3 & 0 & -\varpi_1 \\ -\varpi_2 & \varpi_1 & 0 \end{bmatrix} / \forall \varpi = \begin{bmatrix} \varpi_1 \\ \varpi_2 \\ \varpi_3 \end{bmatrix} \wedge v \\ &= \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \Rightarrow \varpi \times v = \varpi \wedge v.\end{aligned}\quad (48)$$

The main connection between $SE(3)$ and $se(3)$ is the exponential transformation (Equation (49)). It is possible to gen-

eralise the forward kinematic map for an arbitrary 'open-chain' manipulator with n DOF of magnitude $g(\theta)$, through the product of those exponentials, expressed as POE (Equation (50)), where $g(\theta)$ is the reference position for the coordinate system.

$$\left. \begin{aligned}e^{\xi^\wedge \theta} &= \begin{bmatrix} e^{\varpi^\wedge \theta} & (I - e^{\varpi^\wedge \theta})(\varpi \times v) \\ 0 & 1 \end{bmatrix} \in SE(3); \\ e^{\xi^\wedge \theta} &= \begin{bmatrix} I & v\theta \\ 0 & 1 \end{bmatrix} \in SE(3); \varpi = 0, \\ e^{\xi^\wedge \theta} &= I + \varpi^\wedge \sin \theta + \varpi^\wedge 2(1 - \cos \theta),\end{aligned}\right\} \quad (49)$$

$$g(\theta) = \prod_{i=1}^n e^{\xi_i^\wedge \theta_i} \cdot g(0).\quad (50)$$

A very important payoff for the POE formalism is that it provides an elegant formulation for a set of canonical problems, the Paden and Kahan sub-problems (Arbulú et al. 2005a; Pardos and Balaguer 2005), among others, which have a geometric solution for their inverse kinematics. It is possible to obtain a close-form solution for the inverse kinematic problem of complex mechanical systems by reducing them into appropriate canonical sub-problems.

The Paden–Kahan sub-problems are introduced as follows (Murray et al. 1994):

Paden–Kahan 1: rotation about a single axis

Finding the rotation angle using 'screw theory' and Lie groups, at first, the point rotation expression from ' p ' to ' k ' is expressed by (Figure 35)

$$e^{\xi^\wedge \theta} p = k. \quad (51)$$

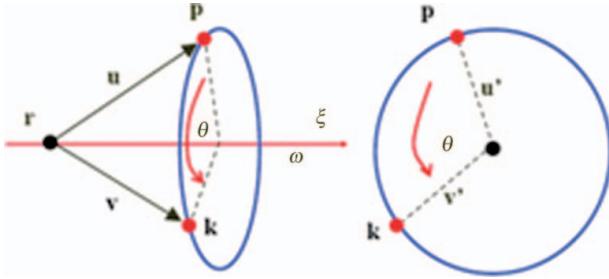


Figure 35. Rotation on single axis ' ω ' from point 'p' to point 'k'.

The twist and projection vectors on the rotation plane are as follows:

$$\xi = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} -\omega \times r \\ \omega \end{bmatrix}, \quad (52)$$

$$u' = u - \omega \omega^T u, \quad (53)$$

$$v' = v - \omega \omega^T v. \quad (54)$$

Finally, the rotation angle is calculated with the following expression:

$$\theta = a \tan 2 [\omega^T (u' \times v'), u'^T \cdot v']. \quad (55)$$

Paden–Kahan 2: rotation about two subsequent axes

The rotation expression is as follows (Figure 36):

$$e^{\xi_1 \wedge \theta_1} e^{\xi_2 \wedge \theta_2} p = e^{\xi \wedge \theta} c = k. \quad (56)$$

The respective twists are described as

$$\xi_1 = \begin{bmatrix} -\omega_1 \times r \\ \omega_1 \end{bmatrix} \wedge \xi_2 = \begin{bmatrix} -\omega_2 \times r \\ \omega_2 \end{bmatrix}. \quad (57)$$

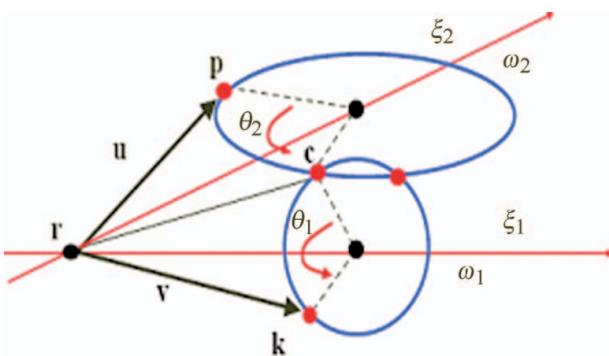


Figure 36. Rotation on two subsequent axes ' ω_1 ' and ' ω_2 ' from p to 'c' and from 'c' to 'k'.

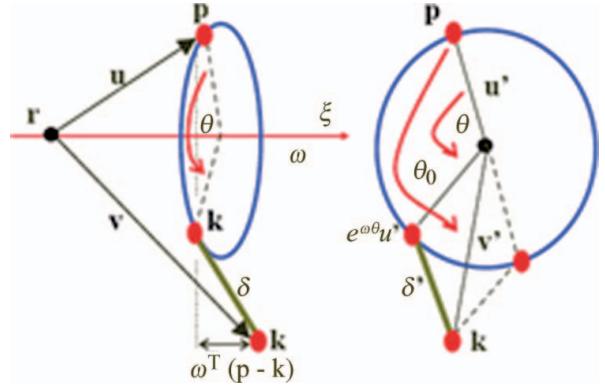


Figure 37. Rotation of point 'p' to 'k', which is a distance ' δ ' from 'q'.

Some values are computed in order to obtain point 'c' with the following expressions:

$$\alpha = \frac{(\omega_1^T \omega_2) \omega_2^T u - \omega_1^T v}{(\omega_1^T \omega_2)^2 - 1}, \quad (58)$$

$$\beta = \frac{(\omega_1^T \omega_2) \omega_1^T v - \omega_2^T u}{(\omega_1^T \omega_2)^2 - 1}, \quad (59)$$

$$\gamma^2 = \frac{\|u\|^2 - \alpha^2 - \beta^2 - 2\alpha\beta\omega_1^T \omega_2}{\|\omega_1 \times \omega_2\|^2}. \quad (60)$$

Obtaining the point 'c'

$$c = r + \alpha \omega_1 + \beta \omega_2 \pm \gamma (\omega_1 \times \omega_2). \quad (61)$$

Once we get 'c' for the second sub-problem, we can apply the first Paden–Kahan sub-problem to obtain solutions for θ_1 and θ_2 . Note that there might be two solutions for 'c', each of them giving a different solution for θ_1 and θ_2 .

Paden–Kahan 3: rotation to a given distance

The distance ' δ ' is shown as follows:

$$\|e^{\xi \wedge \theta} p - q\| = \delta. \quad (62)$$

The associate twist and vectors projection in the perpendicular plane of rotation axis could be computed as

$$\xi = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} -\omega \times r \\ \omega \end{bmatrix}, \quad (63)$$

$$u' = u - \omega \omega^T u, \quad (64)$$

$$v' = v - \omega \omega^T v. \quad (65)$$

Projecting ' δ ' in ' ω ' direction,

$$\delta'^2 = \delta^2 - |\omega^T(p - q)|^2. \quad (66)$$

If we let ' θ_0 ' be the angle between the vectors ' u' ' and ' v' , we have

$$\theta_0 = a \tan 2[\omega^T(u' \times v'), u'^T \cdot v']. \quad (67)$$

Finally, we obtain the rotation angle by

$$\theta = \theta_0 \pm \cos^{-1} \left(\frac{\|u'\|^2 + \|v'\|^2 - \delta'^2}{2\|u'\|\|v'\|} \right). \quad (68)$$

The algorithm developed is called Sagittal Kinematics Division (SKD). It divides the robot body into two independent manipulators, one for the left and the other for the right part of the body (Figure 38), subject to the following constraints at any time: keeping the balance of the humanoid ZMP and imposing the same position and orientation for the common parts (pelvis, thoracic, cervical) of the four humanoid manipulators.

Solving the kinematics problem

It is possible to generalise the leg forward kinematic map with 12 DOF ($\theta_1 \dots \theta_{12}$). The first 6 DOF correspond to the

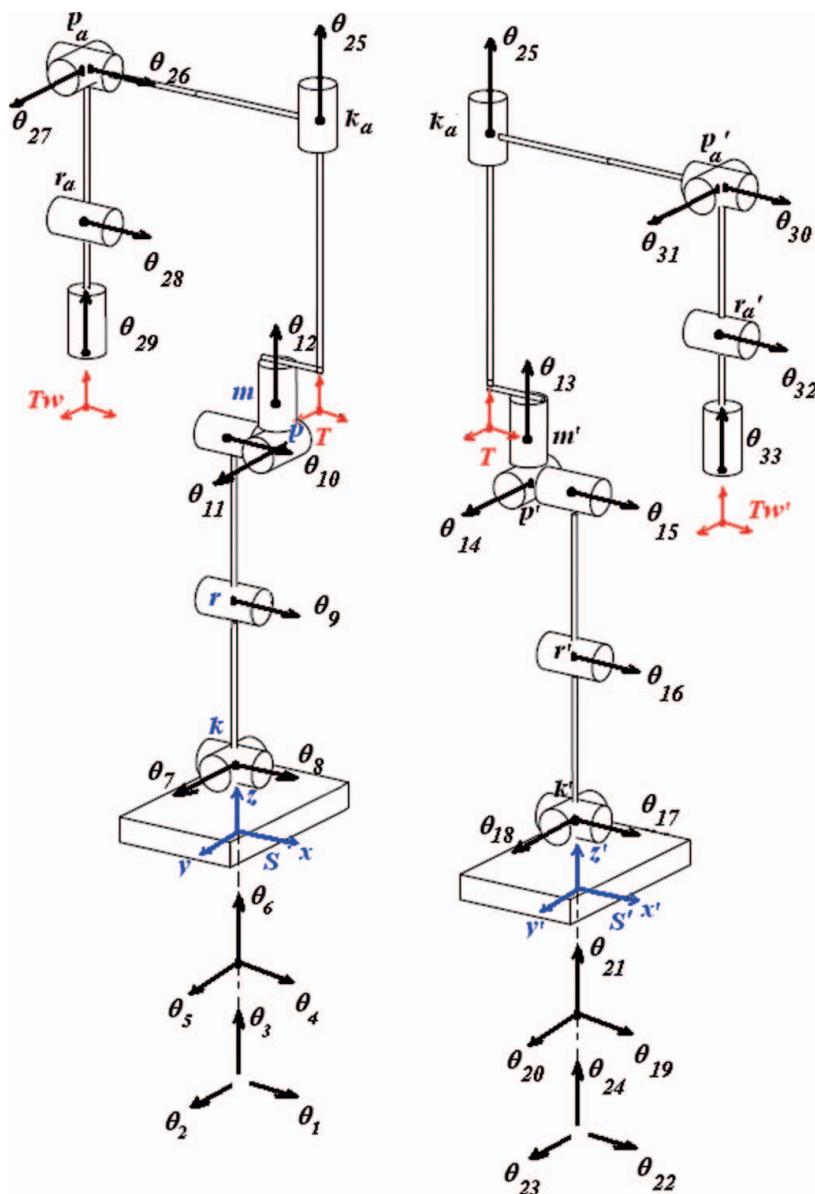


Figure 38. Rh-1 sagittal kinematics division (SKD).

position ($\theta_1, \theta_2, \theta_3$) and orientation ($\theta_4, \theta_5, \theta_6$) of the foot. Note that these DOF do not correspond to any real joint and for that reason we call them ‘non-physical’ DOF.

The other DOF are called ‘physical DOF’ because they correspond to real motorised joints. These are θ_7 for the hind foot, θ_8 for the ankle, θ_9 for the knee, θ_{10} for the hip on the x -axis, θ_{11} for the hip on the y -axis and θ_{12} for the hip on the z -axis. Let S be a frame attached to the base system (support foot) and T be a frame attached to the humanoid hip.

The reference configuration of the manipulator is the one corresponding to $\theta_i = 0$, and $g_{st}(0)$ that represents the rigid body transformation between T and S when the manipulator is at its reference configuration.

Then the product of exponentials formula for the right and left legs’ forward kinematics is $g_{st}(\theta)$ and $g_{s't'}(\theta)$, being ξ^\wedge the 4×4 matrices called ‘twists’.

$$g_{st}(\theta) = e^{\xi_1^\wedge \theta_1} \cdot e^{\xi_2^\wedge \theta_2} \cdots e^{\xi_{29}^\wedge \theta_{29}} \cdot g_{st}(0), \quad (69)$$

$$g_{s't'}(\theta) = e^{\xi_{24}^\wedge \theta_{24}} \cdot e^{\xi_{23}^\wedge \theta_{23}} \cdots e^{\xi_{33}^\wedge \theta_{33}} \cdot g_{s't'}(0). \quad (70)$$

The inverse kinematics problem, i.e. for the right leg (see Figure 38), consists of finding the joint angles, that is the six physical DOF ($\theta_7 \dots \theta_{12}$), given the non-physical DOF ($\theta_1 \dots \theta_6$) from the humanoid footstep planning, the hip orientation and position $g_{st}(\theta)$, achieve the ZMP humanoid desired configuration. Using the POE formula for the forward kinematics it is possible to develop a numerically stable geometric algorithm to solve this problem by using the Paden–Kahan geometric sub-problems. It is straightforward to solve the inverse kinematics problem in an analytical, closed-form and geometrically meaningful way, with the following formulation:

At first, twist and reference configurations are computed.

$$g_{st}(0) = \begin{bmatrix} 1 & 0 & 0 & T_x - S_x \\ 0 & 1 & 0 & T_y - S_y \\ 0 & 0 & 1 & T_z - S_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (71)$$

$$\begin{aligned} v_1 &= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \quad v_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}; \quad v_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; \\ \omega_4 &= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \quad \omega_5 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}; \quad \omega_6 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \end{aligned} \quad (72)$$

$$\begin{aligned} \omega_7 &= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}; \quad \omega_8 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \quad \omega_9 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \end{aligned}$$

$$\omega_{10} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \quad \omega_{11} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}; \quad \omega_{12} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad (73)$$

$$\xi_1 = \begin{bmatrix} v_1 \\ 0 \end{bmatrix}; \quad \xi_2 = \begin{bmatrix} v_2 \\ 0 \end{bmatrix}; \quad \xi_3 = \begin{bmatrix} v_3 \\ 0 \end{bmatrix} \sqrt{2}, \quad (74)$$

$$\xi_4 = \begin{bmatrix} -\omega_4 \times S \\ \omega_4 \end{bmatrix}; \quad \xi_5 = \begin{bmatrix} -\omega_5 \times S \\ \omega_5 \end{bmatrix};$$

$$\xi_6 = \begin{bmatrix} -\omega_6 \times S \\ \omega_6 \end{bmatrix}, \quad (75)$$

$$\xi_7 = \begin{bmatrix} -\omega_7 \times k \\ \omega_7 \end{bmatrix}; \quad \xi_8 = \begin{bmatrix} -\omega_8 \times k \\ \omega_8 \end{bmatrix};$$

$$\xi_9 = \begin{bmatrix} -\omega_9 \times r \\ \omega_9 \end{bmatrix}, \quad (76)$$

$$\xi_{10} = \begin{bmatrix} -\omega_{10} \times p \\ \omega_{10} \end{bmatrix}; \quad \xi_{11} = \begin{bmatrix} -\omega_{11} \times p \\ \omega_{11} \end{bmatrix};$$

$$\xi_{12} = \begin{bmatrix} -\omega_{12} \times p \\ \omega_{12} \end{bmatrix}. \quad (77)$$

Next, it is possible to compute the inverse kinematics as follows: angle θ_9 is obtained using the third Paden–Kahan sub-problem. We pass all known terms to the left side of Equation (69), apply both sides to point p , subtract point k and apply the norm. We operate in such a way because the resulting Equation (78) is only affected by θ_9 , and therefore we can rewrite this equation as Equation (79), which is exactly the formulation of the Paden–Kahan canonical problem for a rotation to a given distance. Thus, we can geometrically obtain the two possible values for variable θ_9 .

$$\begin{aligned} &\|e^{-\xi_6^\wedge \theta_6} \cdots e^{-\xi_1^\wedge \theta_1} g_{st}(\theta) g_{st}(0)^{-1} p - k\| \\ &= \|e^{\xi_7^\wedge \theta_7} \cdots e^{\xi_{12}^\wedge \theta_{12}} p - k\|, \end{aligned} \quad (78)$$

$$\delta = \|e^{\xi_9^\wedge \theta_9} p - k\| \xrightarrow{P-K-3} \theta_9 \text{ Double.} \quad (79)$$

Next, θ_7 and θ_8 are obtained using the second Paden–Kahan sub-problem. We pass all possible terms to the left side of Equation (51) and apply both sides to point p . In doing so, the resulting Equation (80) is only affected by θ_7 , θ_8 and θ_9 , and, therefore, we can rewrite this equation as Equation (81), which is exactly the formulation of the Paden–Kahan canonical problem for two successive rotations.

Therefore, we can geometrically obtain the two possible values, for the pair of variables θ_7 and θ_8 .

$$e^{-\xi_6^\wedge \theta_6} \dots e^{-\xi_1^\wedge \theta_1} g_{st}(\theta) g_{st}(0)^{-1} \\ p = e^{\xi_7^\wedge \theta_7} \dots e^{\xi_{12}^\wedge \theta_{12}} p, \quad (80)$$

$$q' = e^{\xi_7^\wedge \theta_7} e^{\xi_8^\wedge \theta_8} p' \xrightarrow{P-K-2} \theta_7, \theta_8 \text{ Double.} \quad (81)$$

After that, θ_{10} and θ_{11} are obtained using the second Paden–Kahan sub-problem. We pass all known terms to the left side of Equation (69) and apply both sides to point m . As a result of these operations, the transformed Equation (82) is only affected by θ_{10} and θ_{11} , and we can rewrite this equation as Equation (83), which is again the formulation of the Paden–Kahan canonical problem for two successive rotations around crossing axes. Hence, we can geometrically solve the two possible values for the pair of variables θ_{10} and θ_{11} .

$$e^{-\xi_9^\wedge \theta_9} \dots e^{-\xi_1^\wedge \theta_1} g_{st}(\theta) g_{st}(0)^{-1} \\ m = e^{\xi_{10}^\wedge \theta_{10}} e^{\xi_{11}^\wedge \theta_{11}} e^{\xi_{12}^\wedge \theta_{12}} m, \quad (82)$$

$$q'' = e^{\xi_{10}^\wedge \theta_{10}} \cdot e^{\xi_{11}^\wedge \theta_{11}} m \xrightarrow{P-K-2} \theta_{10}, \theta_{11} \text{ Double.} \quad (83)$$

Finally, θ_{12} is obtained using the first Paden–Kahan sub-problem. We pass all known terms to the left side of Equation (69) and apply both sides to point S . As a result, this equation is transformed into Equation (84), which is, obviously, only affected by θ_{12} , and we can rewrite it as Equation (85), a formulation of the Paden–Kahan canonical problem for a rotation around an axis. Thus, we can geometrically obtain the single possible value for variable θ_{12} .

$$e^{-\xi_{11}^\wedge \theta_{11}} \dots e^{-\xi_1^\wedge \theta_1} g_{st}(\theta) g_{st}(0)^{-1} S = e^{\xi_{12}^\wedge \theta_{12}} S, \quad (84)$$

$$q''' = e^{\xi_{12}^\wedge \theta_{12}} S \xrightarrow{P-K-1} \theta_{12} \text{ Single.} \quad (85)$$

The arm motion could be implemented as θ_{25} to θ_{29} solutions. The shoulder and wrist manipulators do not intervene in locomotion and therefore θ_{25} and θ_{29} are zero for the analysed movement. The other arm DOF may or may not contribute to the locomotion, helping the balance control to keep the COG as close as possible to its initial geometric position; but to achieve this behaviour, we must solve the arm inverse dynamic problem, which is beyond the scope of this paper. A very simple but effective practical arm kinematic solution takes advantage of the necessary body sagittal coordination (see the SKD model in Figure 38), and the right arm DOF is made equal or proportional to its complementary left leg DOF. Therefore, the values for the variables θ_{25} to θ_{29} are defined as Equation (86).

$$\theta_{25} = 0; \theta_{26} = \theta_{15}; \theta_{27} = \theta_{14}; \theta_{28} = \theta_{16}; \theta_{29} = 0. \quad (86)$$

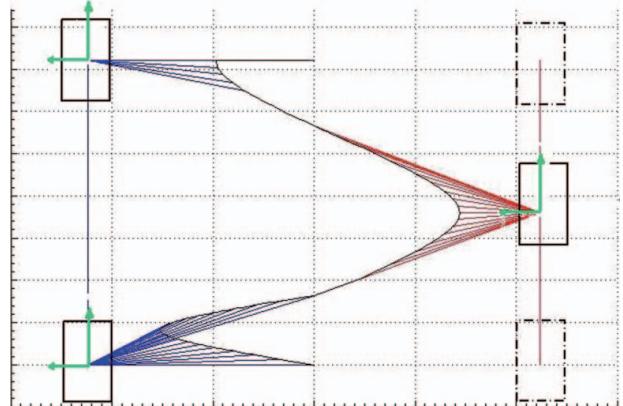


Figure 39. COG hyperbolic trajectories in local axes (green).

With these computations, the right manipulator inverse kinematic problem is solved in a geometric way, and what is more, we have not only one solution but also a set of all possible solutions. For instance, the right leg has eight theoretical solutions, which are captured with the approach shown in Equation (87), if they exist.

$$\begin{aligned} \text{Solutions} &= \theta_9 \text{Double} \times \theta_8 \theta_7 \text{Double} \\ &\times \theta_{11} \theta_{10} \text{Double} \times \theta_{12} \text{Single} = 8. \end{aligned} \quad (87)$$

After repeating exactly the same technique for the left manipulator, the complete Rh-1 humanoid inverse kinematic problem is, in fact, totally resolved.

K. Simulation results

For three steps, Figure 39 shows the spatial motion of the pendulum mass, and the local frame (green frames) of hyperbolic trajectories obtained in the single support phase; the trajectories shape looks like a hyperbolic curve as deduced above. It is a hyperbolic trajectory because the orbital energy in the ‘y-direction’ is negative (this is due to the fact that the pendulum frontal motion accelerates and decelerates without passing the equilibrium point, as shown in Figure 26), so the Equation 39 describes a hyperbole. The passive walkers have another walking principle, which is based on a limit cycle, when the gravity fields act on the device to achieve motion. In our case, we introduce the reference COG motion to make the robot walk, so we can pre-plane the stable walking pattern and introduce it to the humanoid robot. It is noted that the pendulum base is centred in the middle of the support foot and the natural ball pendulum motion follows a hyperbolic trajectory; the established smooth pattern drives the COG of the humanoid robot; natural and stable walking motion is obtained as demonstrated in several simulations and experimental results explained in next paragraphs and sections. Figure 40 shows the temporal

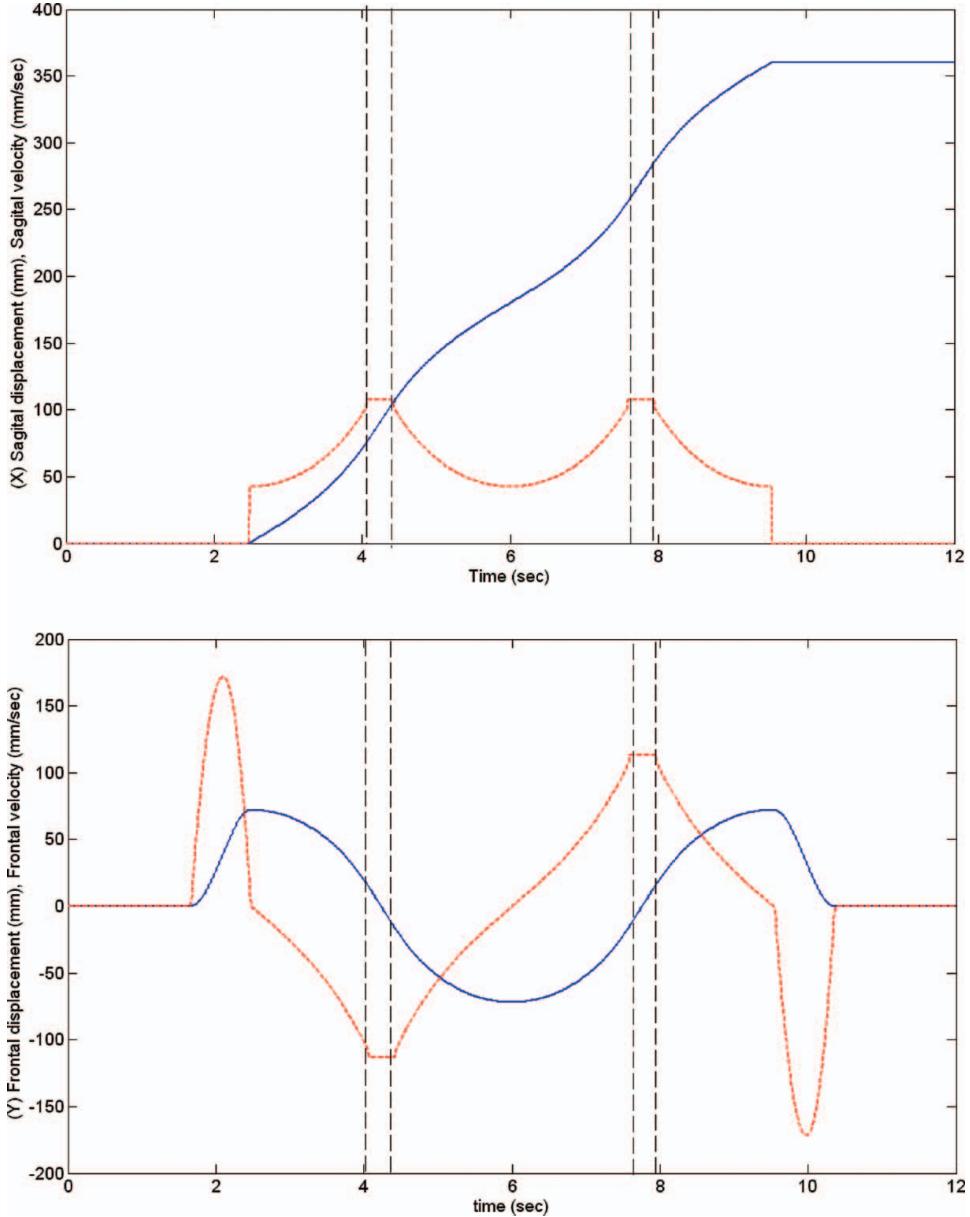


Figure 40. COG temporal position (blue) and velocity (red dotted) patterns for doing three steps. In the double-support phase (between vertical dashed lines), constant speed maintains the trajectory's continuity.

pendulum mass trajectories. In this walking pattern the single support phase takes 1.5 seconds and the double one 0.2 seconds. After computing the inverse kinematics at each local axis (Figure 39), the obtained joint patterns of the right humanoid leg and angular velocities are shown in Figure 41. These allow for checking the joint constraints in order to satisfy the actuator's performances.

Rh-1 simulator results are shown in Figure 42, from the developed VRML environment, which allows us test the walking pattern previously so as to test it in the real humanoid robot. This environment allows us to evaluate

the angle motion range of each joint, avoid self-collision and obstacle collision, in order to obtain adequate walking patterns considering the robot's dimensions and mechanical limitations. It is verified by several simulation tests so that smooth, fast and natural walking motion is obtained using the 3D-LIPM and foot motion patterns.

In order to obtain global humanoid motion for avoiding an external obstacle (Yoshida et al. 2005), walking patterns in any direction could be developed using a rotation matrix around the z -axis of the local frame and the proper boundary conditions such as position and velocity at the local frame of

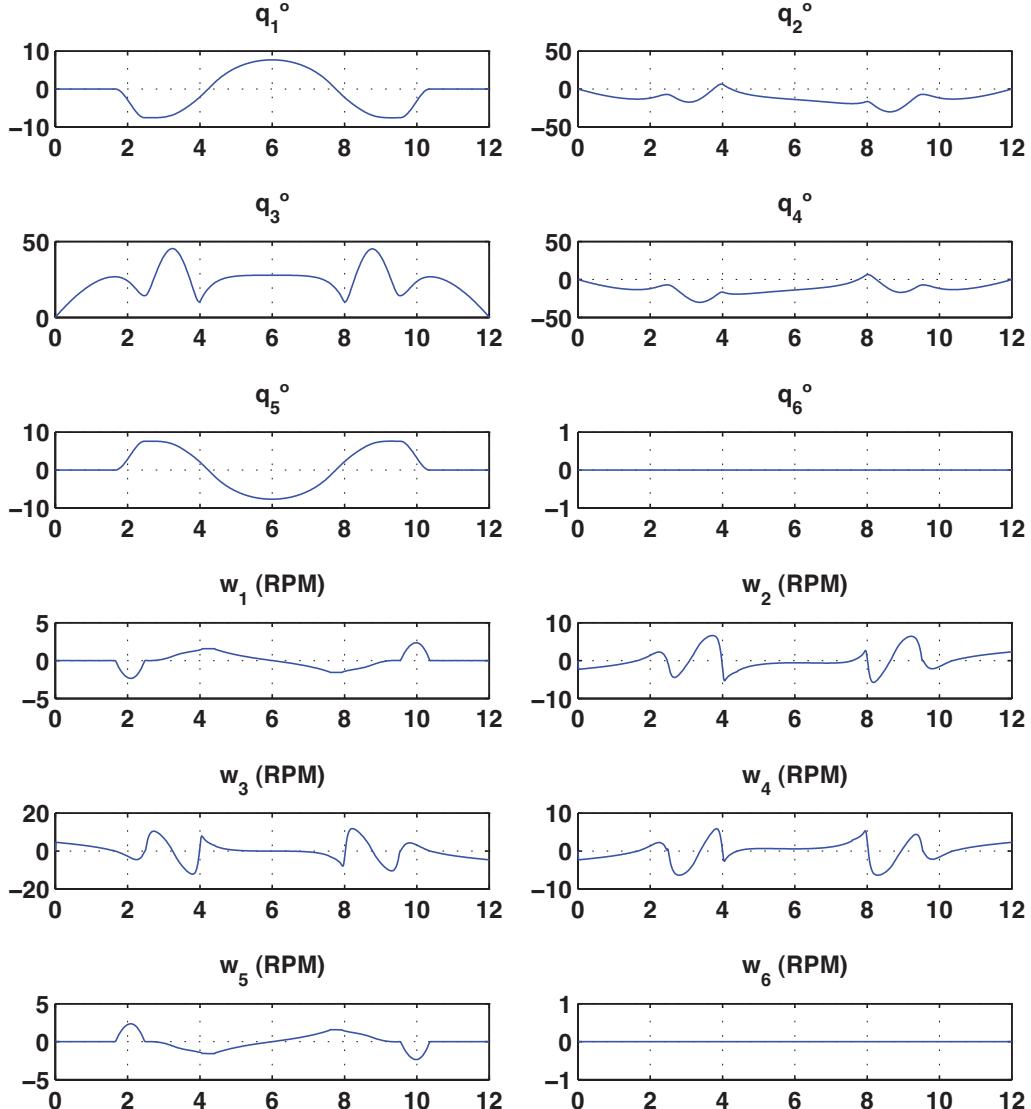


Figure 41. Simulation of joint angle evolution and joint velocity evolution of right leg for three steps walking in a straight line (note that θ_7 , to θ_{12} correspond to q_1 to q_6).

the COG, and foot trajectories while changing the support foot in order to obtain smooth walking patterns. Figures 43 and 44 show us an example of planning walking motion with a change in direction. The walking pattern generated in each local frame maintains continuity with the previous and the next walking pattern. In addition, real-time walking pattern generation is possible, which changes direction, length and step width at any time using the information from sensors or by external command of the humanoid robot, according to the LAG algorithm.

In order to correct mechanical flexion and terrain irregularities, some joint patterns should be modified (i.e. ankle and hip joints) by offline and online control gait. The offline control reduces high joint acceleration at the beginning of

the single support phase because correction starts at this time. Online correction compensates for the actual environment changes and the mechanical imperfections. In this way, stable walking is obtained by maintaining the body's orientation and ZMP in the right position.

VIII. Control architecture

A. Control overview

Humans usually walk using a specific walking pattern in normal environments. In cases of unexpected disturbances, this pattern can be changed immediately to another one in order to adapt to terrain requirements. The humanoid robot

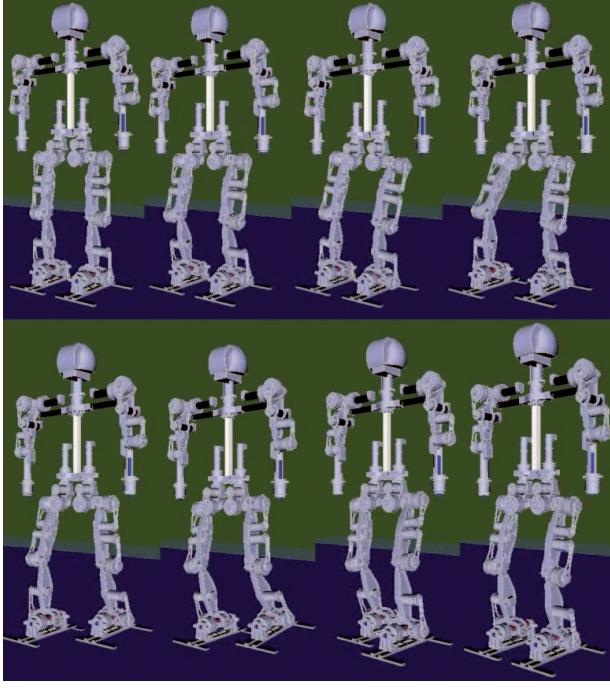


Figure 42. Rh-1 simulator testing walking patterns.

operates in the same space as the human and needs the same mechanism to adept its walking to changing conditions.

Although planned motion patterns satisfy the stability constraints, some errors caused by the irregularity of the terrain or some external forces can cause the humanoid robot to fall down. Moreover, the flexion of the mechanical structure of the humanoid robot is the largest source of errors affecting its walking. To reduce the influence of these errors on walking stability, the control architecture (Figure 45) for online motion patterns' modification has been implemented.

The control architecture consists of two basic parts: Joint Control and Stabilisation Control. The joint control is the core of the control scheme. When the offline-calculated motion pattern is received and the motion is started, the Adaptive Control Algorithm adjusts the motion controller

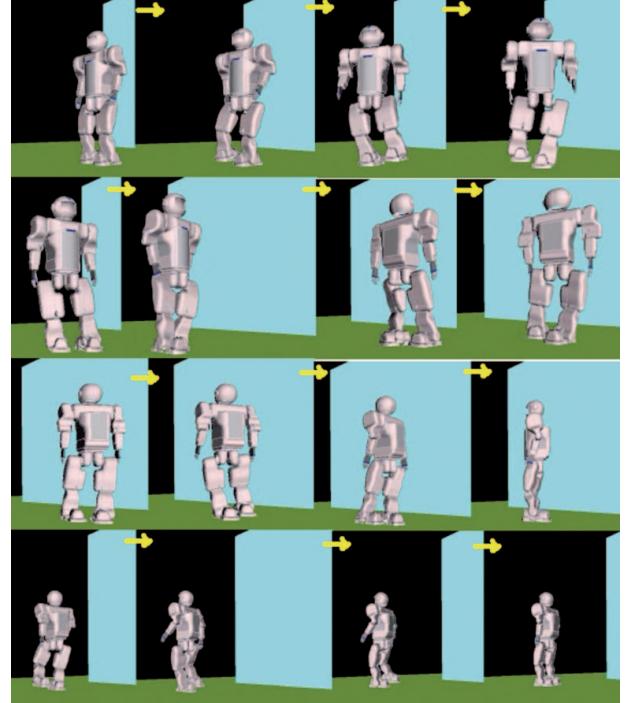


Figure 44. Snapshots of walking patterns in any direction.

of each articulation to the diverse postures of the humanoid robot. If the sensorial system of the humanoid robot detects an error in the body position, stabilisation control corrects it and tries to recover each joint's trajectory in order to execute the previously calculated motion pattern.

When the robot is working in the operational interaction mode (walking, object manipulation etc.), there are several computing and communication tasks that need to be performed in a cyclical mode and be fast enough to avoid any possible loss of control. The periodic (with period T_s) chain (Figure 46) begins with the sensing task, taking the time interval t_{att} for attitude estimation gyros and accelerometer readings and t_{zmp} for ZMP force-torque sensor readings.

These tasks are followed by the tasks performing CAN bus communications, posture control and stabiliser and inverse kinematic calculator computing, internal PC bus communications, supervisory controller and CAN bus transmission of new reference for each joint of the humanoid robot.

The period T_s should be small and compatible with the dynamics of the humanoid robot movement. On the one hand, T_s cannot be made arbitrarily small because the various computing and communication tasks with execution times t_{att} , t_{zmp} , t_{CAN} , t_{PC} , $t_{Stabilizer}$ etc., cannot themselves be made arbitrarily small. Also, a small value for T_s would generate too many messages in communication lines (PC bus, CAN bus) that would overload it. On the other hand, T_s cannot be made arbitrarily large because of the dynamics of the robot (Nyquist criterion).

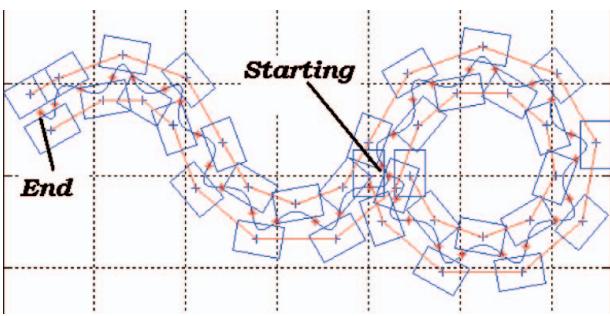


Figure 43. Generating walking patterns in any direction.

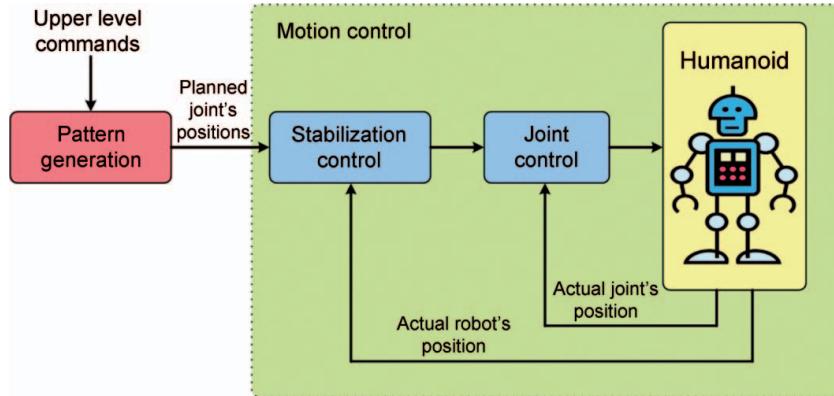


Figure 45. Control architecture.

Thus,

$$Ts > t_{att} + t_{zmp} + t_{CAN} + t_{PControl} + t_{Stabilizer} + t_{InvKin} + t_{PC} + t_{Sup} + t_{CAN} \quad (88)$$

and

$$Ts < \frac{1}{2 \cdot Fr}, \quad (89)$$

where Fr is the highest movement frequency on any robot link.

Assuming that the robot walks at about the same speed as a human walking normally, $Fr = 2$ Hz.

Thus,

$$Ts < 250 \text{ ms}. \quad (90)$$

This upper limit of the sample time is a strict real-time requirement considering the complexity of the computing and communication tasks to be performed within this time limit.

The following sections present the detailed design of joint and stabilisation controls.

B. Joint control

Different postures of the humanoid robot radically change the motors' dynamics and require adaptive algorithms to improve the control of each joint. The main goal is to achieve adequate system response for the desired humanoid robot movements.

In the humanoid robot Rh-1, each motor is controlled by the motion controller. This motion controller includes the bottom three cascade control loops (Figure 47): position loop (proportional), velocity loop (proportional–integral) and current loop (proportional–integral).

This type of control scheme has a reasonable advantage over a simple loop scheme. The internal, faster loop absorbs the major part of disturbances until these affect the more important external loop.

Moreover, the adaptive control algorithm is considered in order to maintain the real trajectory of each joint as similar as possible to the ideal one (motion pattern), obtained by solving the kinematic problem.

Equation (73) presents a classical dynamic model of a loaded DC motor, which is usually used to move the joints of a humanoid robot

$$\Omega(s) = \frac{Ks}{(D + s \cdot J)} \cdot I_a(s) - \frac{1}{(D + s \cdot J)} \cdot T_L(s), \quad (91)$$

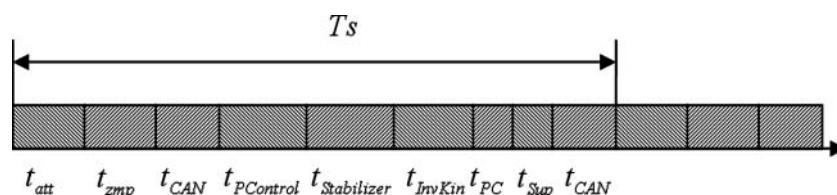


Figure 46. Main computing and communication tasks for Rh-1.

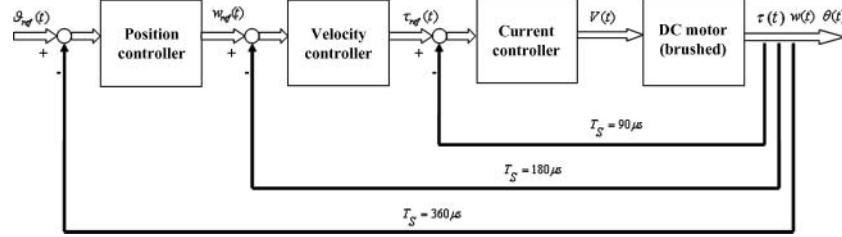


Figure 47. Block diagram of implemented motion controller.

where $\Omega(s)$ = velocity of rotor of the DC motor, $I_a(s)$ = current, K_s = velocity constant, D = viscous friction or viscous damping inherent in the system, J = total moment of inertia of a joint and $T_L(s)$ = load torque. Evidently, when a humanoid robot moves, it assumes different postures characterised by different configurations of joints. This leads to continuous changing in the model of a DC motor. Thus, the adaptive algorithm, tuning the controller of each joint when the model of a motor changes, is required. It carries out online parameter modification of the motion controller in compliance with the system's behaviours.

The most suitable way to implement an adaptive control scheme is the gain-scheduling control. It does not need online identification. The offline software integrating identification and controller design algorithms estimate position and velocity controllers' gains for every possible configuration of the humanoid robot. The identification is made for real gaits (forward walking, backward walking and turning) in order to consider the influence of the posture of the robot on the system dynamics at every moment of the motion. In other words, every offline trajectory (gait) is divided into small parts. For each part, its own values of controller's gains are estimated and a multidimensional table of gains is

created. Figure 48 shows the block diagram of the designed gain-scheduling motion controller.

The online part of the adaptive algorithm updates the controller's parameters following the offline-calculated table. In general, it is necessary to choose adequate parameters according to the actual angular position of each joint and the posture of the whole robot.

This control approach has some advantages in comparison with the online adaptive control algorithms. It does not require a lot of online computational resources, which is a critical restriction for implementing the controller for each of the 21 joints of the humanoid robot Rh-1.

C. Stabilisation control

As was shown in the previous section, the humanoid robot's dynamics is governed by the ZMP Equation (45). From these equations it can be concluded that there are two main variables to control:

$$e_{ZMP} = ZMP^d - ZMP^a, \quad (92)$$

$$e_{COG} = COG^d - COG^a, \quad (93)$$

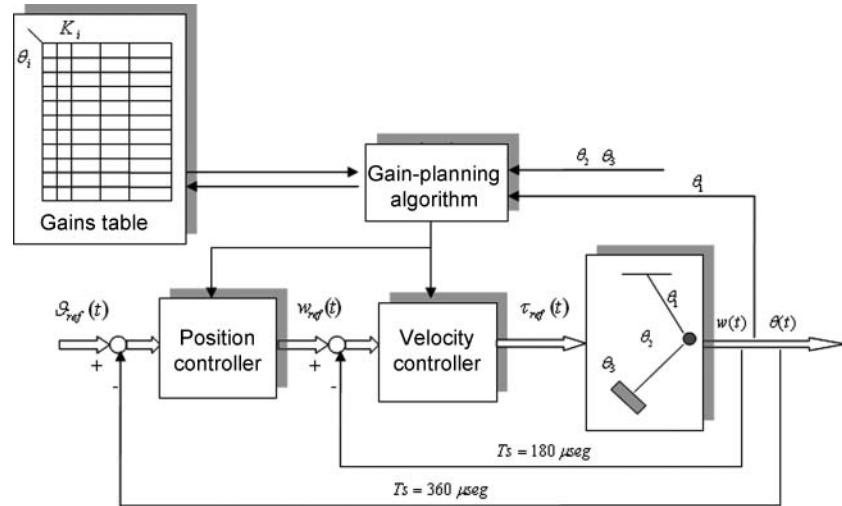


Figure 48. Block diagram of gain-scheduling motion controller.

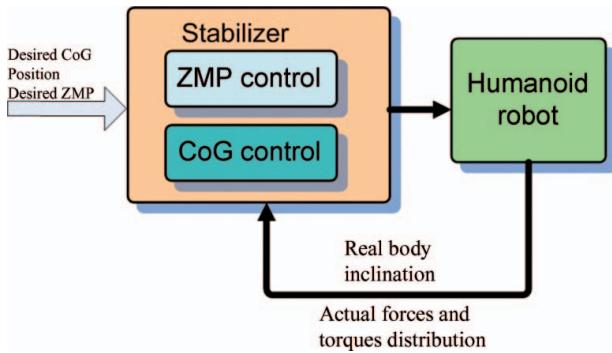


Figure 49. Stabiliser structure.

where ZMP error is denoted as e_{ZMP} and COG position error is denoted as e_{COG} . ZMP^d and COG^d are desired and ZMP^a and COG^a are actual ZMP and COG positions, respectively. Equations (74) and (75) state that two different controls should be implemented in order to stabilise humanoid robot walking. Figure 49 presents the main structure of the stabilisation controller for the humanoid robot.

Previously computed desired COG and ZMP positions generate joint trajectories, which are sent for the humanoid motion. During the execution of motion patterns, the real ZMP and COG positions should be measured. The stabiliser compares it with the ideal ones and generates a new corrected motion pattern for the next moment. This control scheme provides a simple and effective way of controlling humanoid walking stability.

D. ZMP control

The implementation of the ZMP control should provide an easy and robust method to move the actual ZMP to the point where the target ZMP is located. The easiest way here is

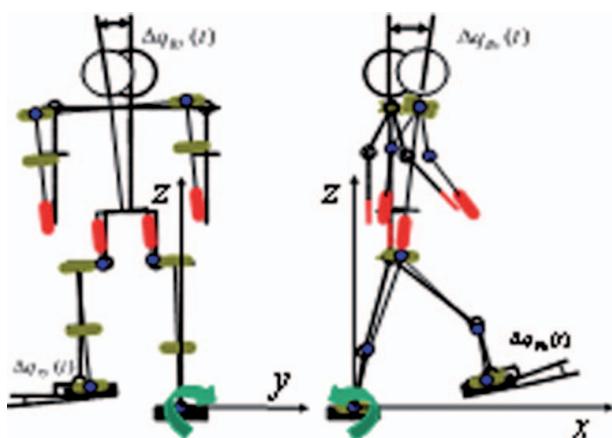


Figure 50. ZMP error compensation.

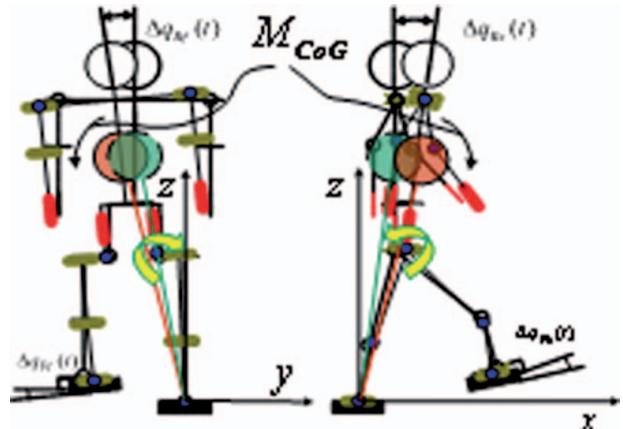


Figure 51. COG error compensation – attitude control.

to neglect the influence of the part of the humanoid robot above its ankle joint. It can be made by the rotation of the upper part of the mechanism with respect to the ankle joint as shown in Figure 50.

Although the proposed compensational mechanism allows the actual ZMP to be moved to the point where the target ZMP is located, some problems appear. When the upper body is rotated with respect to the ankle joint, body angle errors in frontal $\Delta q_{Bf}(t)$ and sagittal $\Delta q_{Bs}(t)$ planes appear (Figure 50).

These errors in frontal and sagittal planes cause the appearance of tilting torques and errors in the frontal $\Delta q_{Ff}(t)$ and sagittal $\Delta q_{Fs}(t)$ planes during the positioning of the hanging foot. It can overturn the robot or introduce very strong instability and vibrations when the foot is landing. Therefore, the mechanism for compensating for the body inclination errors in the frontal $\Delta q_{Bf}(t)$ and sagittal $\Delta q_{Bs}(t)$ planes should complement the ZMP control.

E. COG – attitude control

Using a 3D-LIPM (or other method) for trajectory generation we assume that the humanoid's upper body does not have any influence on the dynamics of a model. In the real walking case it leads to unbounded errors in the COG position. It differs radically from the ideal position computed for a motion pattern. The tilting moment M_{CoG} compounded by M_x and M_y components denotes the robot's upper body dynamics (Figure 51) and should be compensated by a control algorithm. The Attitude control is a special sort of a COG control, which tries to maintain the trunk of the robot strictly vertical in every stage of its motion, thus eliminating the tilting moment and body inclination errors.

The attitude of the robot is normally determined by a combination of gyroscopes and accelerometers placed in the waist, where the approximate centre of the mass or

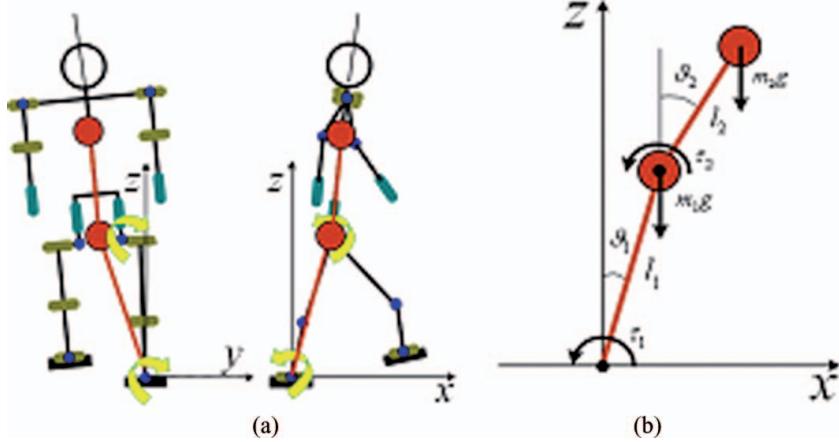


Figure 52. (a) Humanoid robot motion modelling; (b) double inverted pendulum.

the upper part of its backbone is located. As soon as the sensorial information is obtained, the most effective way to control the humanoid robot's body inclination is to maintain its backbone strictly vertical at every stage of the trajectory. In this case, it is enough to control only the hip joints in the frontal and sagittal planes.

F. Double inverted pendulum

As has been shown in the previous sections, to maintain the stability of a humanoid robot it is necessary to implement simultaneously both the ZMP and the Attitude controls acting on the ankle and hip joints. The humanoid robot in that case should be modelled as an inverted double pendulum as shown in Figure 52(a).

A double pendulum consists of one pendulum attached to another. Consider a double bob pendulum with masses m_1 and m_2 ($m_1 + m_2 = M$ – the total mass of the humanoid robot) attached by rigid massless wires of lengths l_1 and l_2 . Furthermore, let the angles that the two wires make with the vertical be denoted as ϑ_1 (ankle rotation) and ϑ_2 (hip rotation), as illustrated in Figure 52(b). The position of the centre of mass of two rods may be written in terms of these angles. The potential energy of the system is given by

$$V = (m_1 + m_2)gl_1 \cos \vartheta_1 + m_2gl_2 \cos \vartheta_2. \quad (94)$$

The kinetic energy of the system is given by

$$\begin{aligned} T &= \frac{1}{2}m_1l_1^2\dot{\vartheta}_1^2 \\ &+ \frac{1}{2}m_2[l_1^2\dot{\vartheta}_1^2 + l_2^2\dot{\vartheta}_2^2 + 2l_1l_2\dot{\vartheta}_1\dot{\vartheta}_2 \cos(\vartheta_1 - \vartheta_2)]. \end{aligned} \quad (95)$$

Writing the Lagrangian of a system using Equations (94) and (95) for ϑ_1 , ϑ_2 and simplifying, we get

$$\begin{aligned} &(m_1 + m_2)l_1^2\ddot{\vartheta}_1 + m_2l_1l_2\ddot{\vartheta}_2 \cos(\vartheta_1 - \vartheta_2) + \\ &+ m_2l_1l_2\dot{\vartheta}_2^2 \sin(\vartheta_1 - \vartheta_2) - l_1g(m_1 + m_2)\sin\vartheta_1 \\ &= \tau_1, \end{aligned} \quad (96)$$

$$\begin{aligned} &m_2l_2^2\ddot{\vartheta}_2 + m_2l_1l_2\ddot{\vartheta}_1 \cos(\vartheta_1 - \vartheta_2) - \\ &- m_2l_1l_2\dot{\vartheta}_1^2 \sin(\vartheta_1 - \vartheta_2) - l_2m_2g\sin\vartheta_2 = \tau_2. \end{aligned} \quad (97)$$

The aim of the present research is to develop a simple control strategy allowing for the use of traditional controllers such as the linear quadratic regulator (LQR) or a simple PID controller to regulate the inverted pendulum about the upright equilibrium point. As the name may suggest, the LQR controller requires a linear system for which it will generate constant gains for full state feedback to make the equilibrium point globally asymptotically stable. However, the dynamics of double inverted pendulum systems are inherently non-linear. The approach chosen was to linearise the equations of motion regarding the operating point and define a domain of attraction within which the constant gain controller results in local asymptotic stability.

Take the approximation that perturbations in the problem are very small, and those terms of second and higher orders are negligible. In this limit $\cos \vartheta_i = 1$, $\sin \vartheta_i = \vartheta_i$ and $\cos(\vartheta_1 - \vartheta_2) = 1$. Applying a small angle approximation we get the following linearisation:

$$(m_1 + m_2)l_1^2\ddot{\vartheta}_1 + m_2l_1l_2\ddot{\vartheta}_2 - l_1g(m_1 + m_2)\vartheta_1 = \tau_1, \quad (98)$$

$$m_2l_2^2\ddot{\vartheta}_2 + m_2l_1l_2\ddot{\vartheta}_1 - l_2m_2g\vartheta_2 = \tau_2. \quad (99)$$

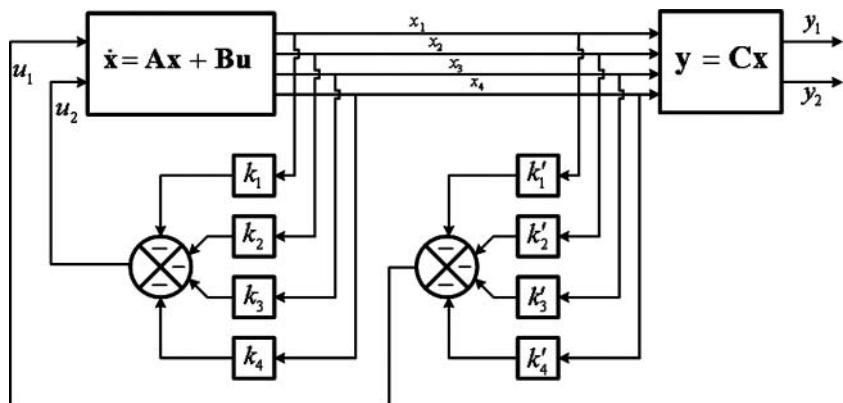


Figure 53. Double inverted pendulum control system.

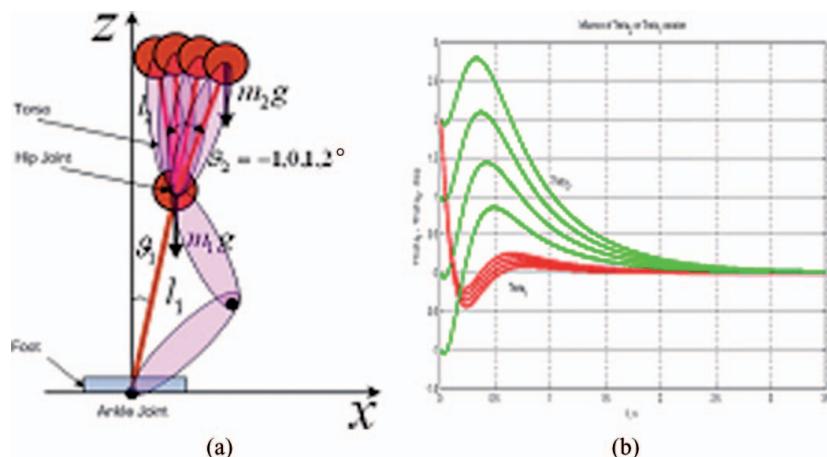


Figure 54. (a) Double inverted pendulum ϑ_2 variation; (b) influence of the ϑ_2 on ϑ_1 variation.

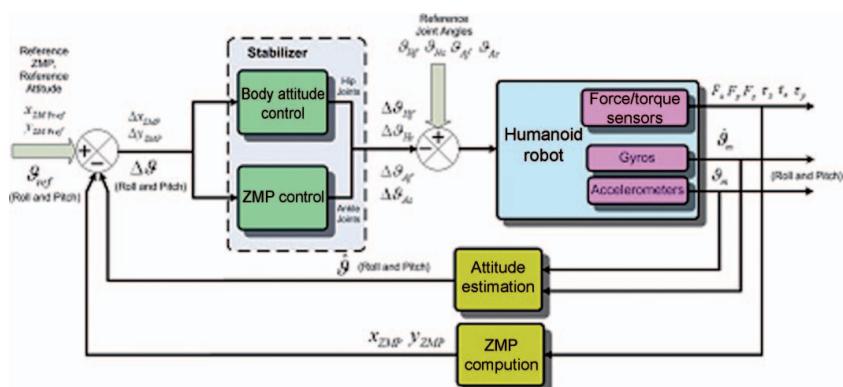


Figure 55. Stabiliser architecture.

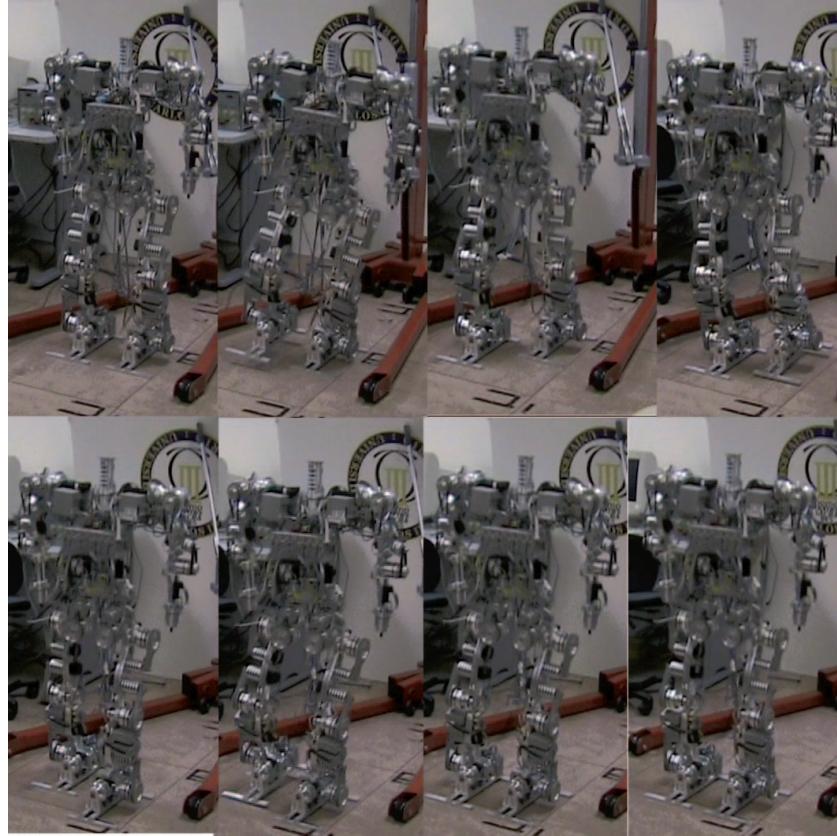


Figure 56. Snapshots of the actual Rh-1 humanoid robot walking with dynamic gait.

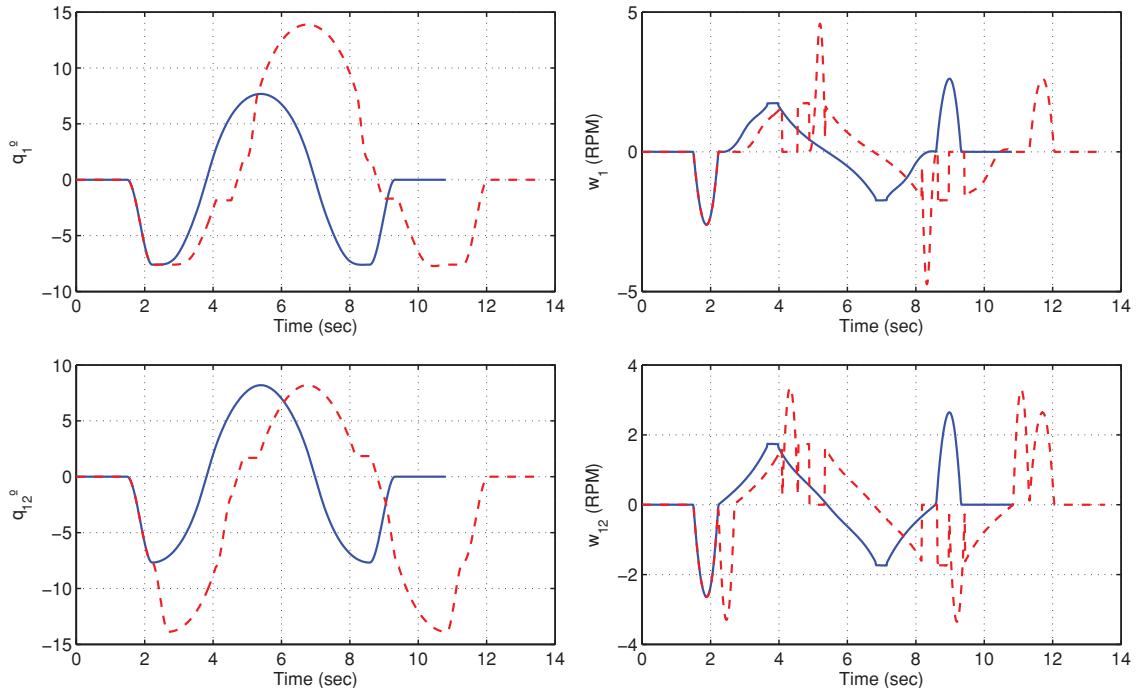


Figure 57. The ankle joint's frontal motion (angular and velocity evolution), original evolution (blue line), compensated evolution (red dotted line).

The state representation of the dynamical system in standard form is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{g(m_1 + m_2)}{m_1 l_1} & 0 & -\frac{m_2 g}{m_1 l_1} & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{g(m_1 + m_2)}{m_1 l_2} & 0 & \frac{g(m_1 + m_2)}{m_1 l_2} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{m_1 l_1^2} & -\frac{1}{m_1 l_1 l_2} \\ 0 & 0 \\ -\frac{1}{m_1 l_1 l_2} & \frac{(m_1 + m_2)}{m_1 m_2 l_2^2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (100)$$

With the output equation

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad (101)$$

where state variables x_1 , x_2 , x_3 and x_4 are defined as $x_1 = \vartheta_1$, $x_2 = \dot{\vartheta}_1$, $x_3 = \vartheta_2$ and $x_4 = \dot{\vartheta}_2$.

LQR controller design considers an optimal control problem that for given system of Equations (100) and (101) determines the matrix K of the optimal control vector:

$$u(t) = -Kx(t). \quad (102)$$

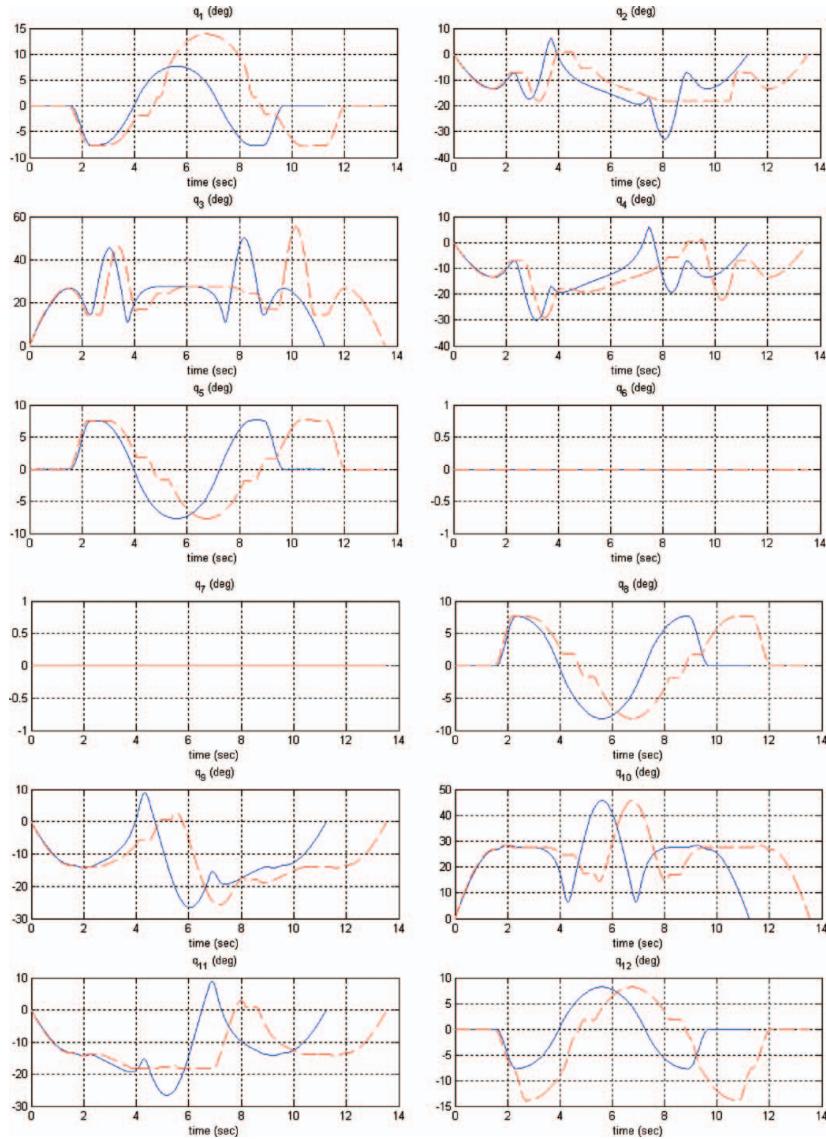


Figure 58. Both leg joint patterns. Reference (blue) and offline corrected (red dotted) overlapped.

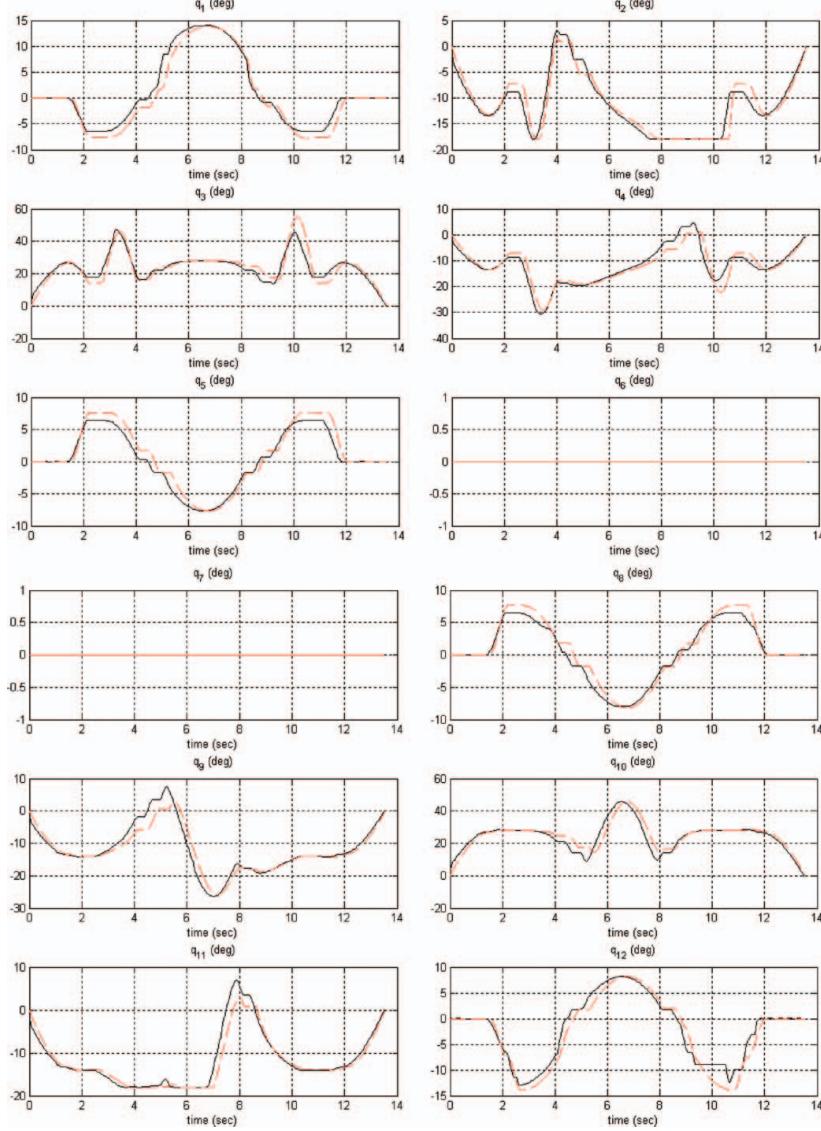


Figure 59. Both leg joint patterns. (a) Reference (blue) and offline corrected (red dotted) overlapped; (b) offline corrected and measured (black) overlapped.

The block diagram showing the optimal controller configuration for the double inverted pendulum system is presented in Figure 53.

Note that the control system presented in Figure 46 is the regulator system. In this case the controller maintains desired angles ϑ_1 and ϑ_2 of the double pendulum close to zero.

G. Decoupled control

Although the stabilisation method for the humanoid robot based on the double inverted pendulum dynamics presented in the previous section seems to be the most appropriate

one for controlling the mechanism, in practice it has many drawbacks. The main reason is that the double inverted pendulum considers control torques of both (hip and ankle) joints of the humanoid robot involved in the stabilisation control. This means that it is necessary to use the torque-controlled actuators to implement this control algorithm. Most contemporary humanoid robots are driven by DC motors with position control where torque control cannot be implemented. A decoupled approach considers that ZMP and Attitude can be implemented as totally independent and the influence of one on another will be negligibly small. To prove the appropriateness of this, let us examine the functioning of the already developed double inverted pendulum

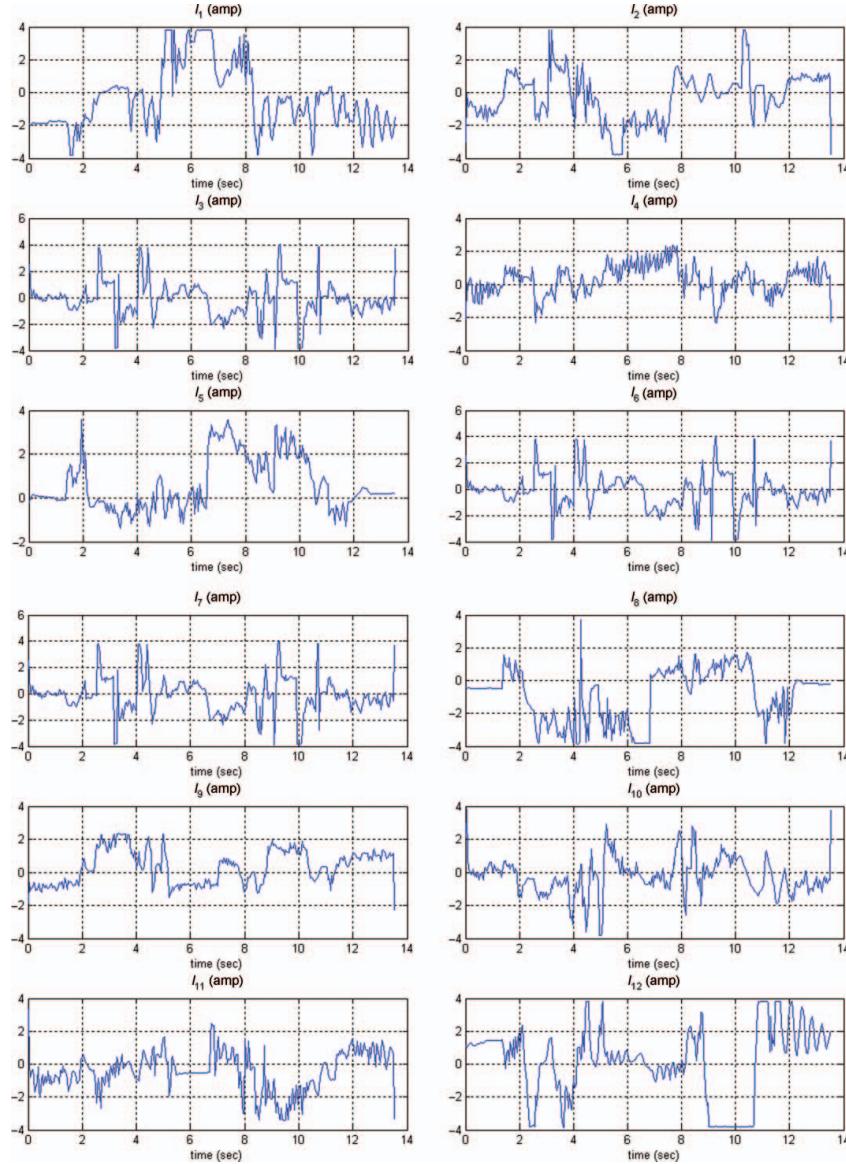


Figure 60. Both legs' current consumption.

system in diverse initial positions of the pendulum (Figure 54), which correspond to different perturbations acting on the humanoid robot:

From Figure 54(b) it can be observed that a large variation in ϑ_2 does not have much influence on ϑ_1 . Its variations practically stay at the same level in every experiment. Therefore, we take into consideration that the control of ϑ_2 will have more priority (or gain) and thus ϑ_2 variation will be practically zero. On the other hand, variation of ϑ_1 sensed by ϑ_2 can be considered as an additional perturbation and will be compensated for by the controller. Moreover, it should be noted that the dynamics of the real robot would be attenuated by the reduction between its links. Thus, stabiliser control of the humanoid robot can be considered as

a sum of two decoupled components related to posture control (ϑ_2) and ZMP control (ϑ_1). In practice, decoupled ZMP and Attitude controllers were designed as LQR controllers.

After the foregoing detailed development of all the parts, we now turn to the detailed control architecture (see Figure 55).

A sensorial system of the robot consisting of two six-axis force-torque should provide the controller with the real distribution of forces and torques F_x , F_y , F_z , τ_x , τ_y , τ_z at the contact point of the foot with the ground. The three-axis Gyro and accelerometer provide the measurements of angular position ϑ_m and angular velocity $\dot{\vartheta}_m$ of the upper body (trunk) of the robot in the frontal and sagittal planes (roll and pitch), (Baerveldt and Klang 1997; Löffler et al.

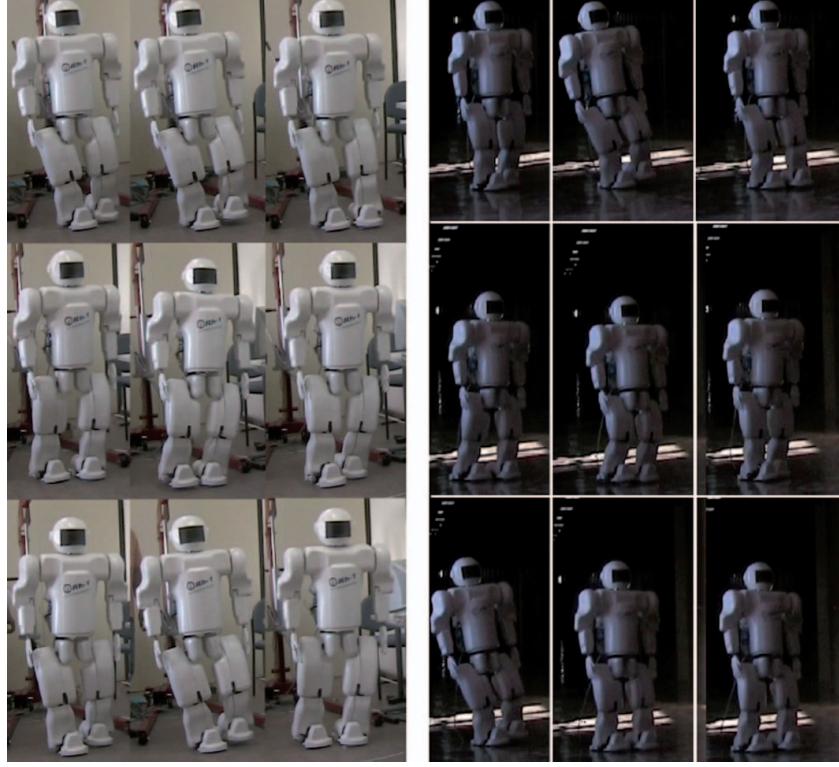


Figure 61. Rh-1 snapshots walking forward.

2003). After the actual ZMP positions x_{ZMP} and y_{ZMP} are computed by the ZMP computational module and the real attitude is estimated in the attitude estimation module ZMP Δx_{ZMP} , Δy_{ZMP} and Attitude $\Delta\vartheta$, errors can be estimated. These errors are the input data for the stabiliser. The stabiliser is designed as a decoupled controller. It controls error in ZMP and Attitude positioning of the humanoid robot by the motion of the ankle and hip joints as discussed above. Finally, the compensational motion of the ankle $\Delta\vartheta_{Af}$, $\Delta\vartheta_{As}$ and hip $\Delta\vartheta_{Hf}$, $\Delta\vartheta_{Hs}$ joints in the frontal and sagittal planes should be superimposed with their reference trajectories ϑ_{Af} , ϑ_{As} , ϑ_{Hf} and ϑ_{Hs} which are responsible for the entire walking process. By doing this, new motion patterns for ankle and hip joints are generated. The implementation of the decoupled stabiliser provides fast and easy control of the walking stability of the humanoid. All changes are applied to the ankle and hip joints eliminating the need for inverse kinematics computation.

IX. Experimental results

Once the design, construction and assembly stages were carried out, the robot's experimental results followed.

Experimental results applying the natural 3D-LIPM and Cart-table models with Body Posture Control algorithms on the Rh-1 humanoid robot platform are discussed in this section. At the beginning, snapshots of dynamic walking of

the Rh-1 humanoid robot are shown in Figure 48. As shown in the VRML (Figure 42) environment, the same walking motion pattern is followed.

The gait patterns proposed above allow for stable walking at 1 km/h. Smooth and natural walking motions are obtained by introducing the adequate initial conditions on COG body motion and correct boundary conditions in foot planning. Some results are shown in Figure 56, where the frontal ankle joint's evolution can be seen in the chart, i.e. q_1 is the frontal right ankle joint, the blue line is the pattern obtained from the inverse kinematics model using the COG and foot patterns as inputs, for doing three steps; the red line is the on-line compensated reference pattern which allows for frontal humanoid body reorientation (see Figure 57). This compensation is necessary because the mechanical elasticity tilts the humanoid into an unstable zone. On the other hand, w_1 is the frontal joint angular velocity of the right ankle, the blue line is the velocity pattern obtained without compensation, the red one is the compensated velocity; adequate interpolators have been developed in order to maintain the humanoid body in the stable zone. Compensated curves are delayed with respect to reference ones, because there exists a compensation time which starts in the single support phase. The q_{12} and w_{12} patterns correspond to the left ankle frontal joint. In the sagittal motion, a similar compensation has been made.

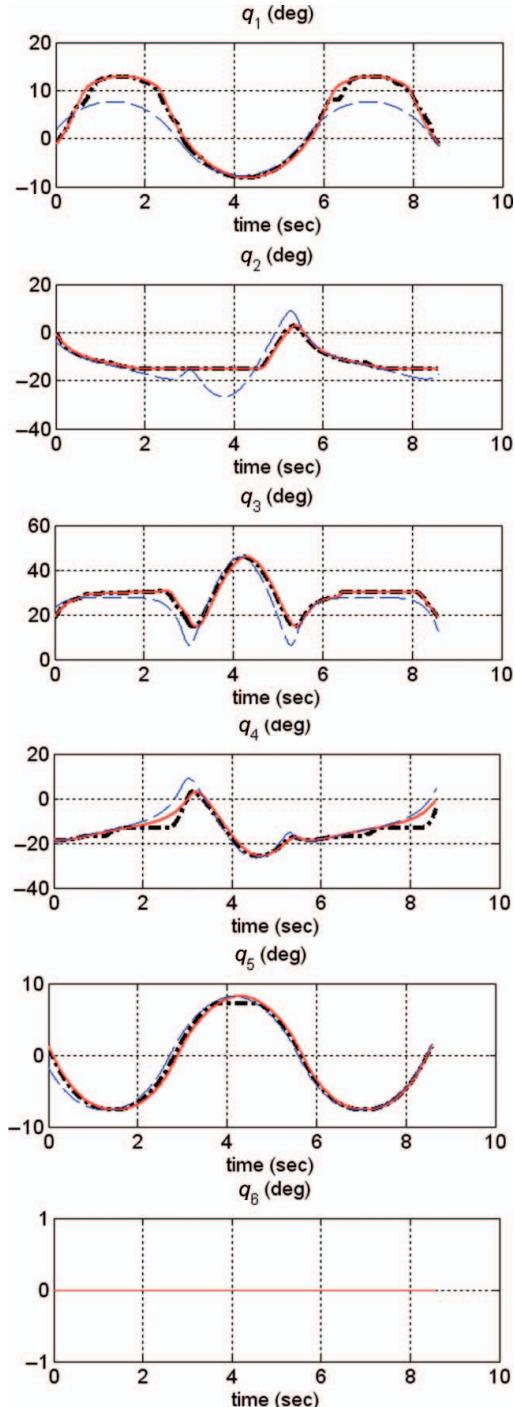


Figure 62. Real joint angular evolution (dot dashed black line), offline compensated (continuous red line) and reference (blue dashed line). Right leg.

This compensation maintains the COG following pendulum laws, and thus, natural and stable walking is obtained. Posture compensation allows for a stable gait because the natural 3D-LIPM spatial trajectories are followed. Offline and online posture control compensates for several

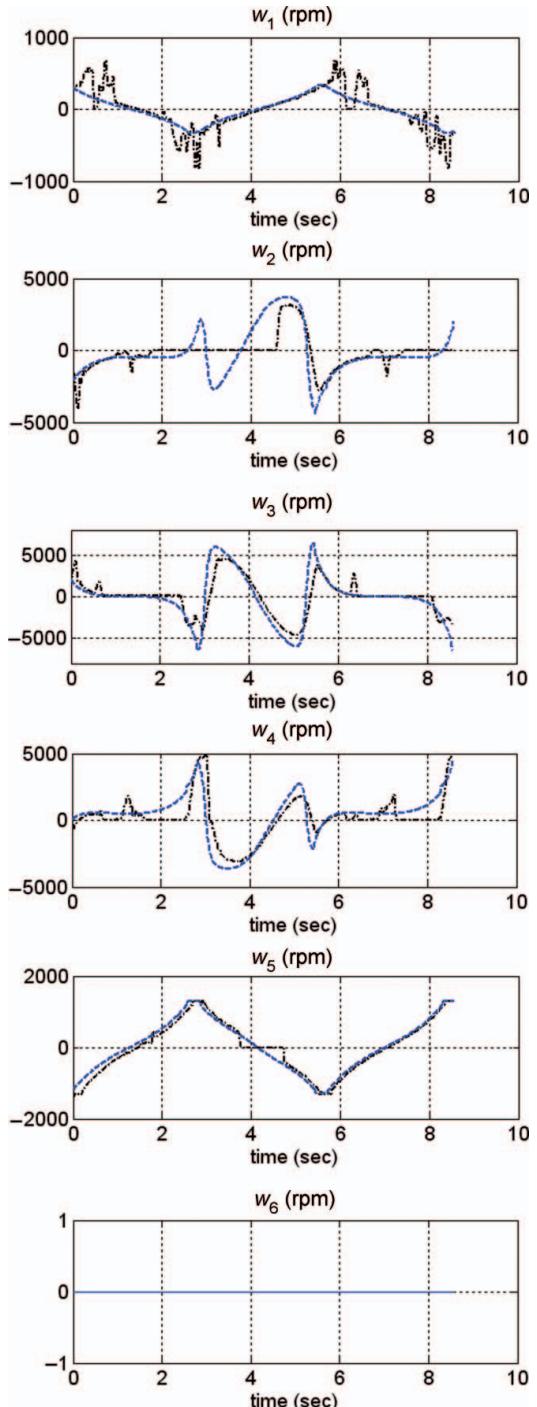


Figure 63. Real (dot dashed black line) and reference (blue dashed line) motor angular velocity evolution. Right leg.

things: the gravitational effect on the humanoid structure in the sagittal and frontal motion planes, which tilts the humanoid to an unstable position; and terrain irregularities and errors in the concentrated mass model that we have used for generating walking patterns. The offline-compensated and real measured joint pattern references for both legs are

shown in Figures 58 and 59; furthermore the joints' current consumptions are presented in Figure 60, in order to check the actuators current limits. Next, snapshots following the Cart-table model are shown in Figure 61. Successful results of straight walking motion in the laboratory and hall have been obtained. The step length is about 180 mm, max feet height is 40 mm and step time about 1.25 sec. The joint patterns and angular velocities of the right leg results are shown in Figures 62 and 63. The reference, offline-corrected and real measure patterns of each joint for taking three steps forward are overlapped. The main differences are overlapped between the offline compensation and measured joint evolution because of the variations, friction on the surface and mechanical imperfections of the robot.

We test that the control system allows for a stable walking motion in a straight line. Further improvements on the mechanical structure, walking pattern generation and control (i.e. foot landing with compliance control, for reducing the impact forces) in order to compensate for whole body moments, will induce a smooth and natural walking motion.

X. Conclusions

Normal bipedal gait is achieved through a complex combination of automatic and volitional postural components. Normal walking requires stability to provide antigravity support of body weight, mobility of body segments and motor control to sequence multiple segments while transferring body weight from one limb to another. The result is energy-efficient forward progression. The human 'gait cycle' has been analysed in order to understand biped walking motion in its main phases, single support and double support phases and their properties: force reaction, cycle time, foot, knee, hip and body motion trajectories. In this way humanoid robot trajectories can be created as that of human ones. It is demonstrated that the COG human motion follows the inverted pendulum laws at normal walking velocity, which is an important fact for maintaining stability while walking.

Concerning the facts previously explained, it is possible to state that very satisfactory results were obtained, thus being a starting point for innumerable investigations in the future.

At the moment, many improvements and corrections are to be done to the mechanical parts. Due to great amount of elements working together, some unwanted clearances and movements in the mechanical structure of the robot may appear. Furthermore, the robot is in its second-evaluation stage and the number of elements that make up the robot must be decreased, either by redesigning the most critical ones or by fusing several of them into one. Compliance foot improvements will be implemented in order to reduce the efforts on each joint and overall structure.

Considering the hardware and software architecture of the Rh-1 robot, we must point out that this work makes an

effort to show that there is a possibility of bringing some basic aspects of industrial automation and control to the other, more sophisticated fields of robotics, in order to extend further standardisation and unification of the design processes. Moreover, the proposed approach allows for consideration of humanoid robot locomotion inside the global automation problem.

Dynamic walking was successfully implemented in the Rh-1 humanoid robot. It can walk smoothly and about 20 times faster than when using a static walking pattern, as was studied in previous works.

The SKD humanoid model makes it easy to solve the inverse kinematic problem using Lie groups' math techniques, such as the POE. For bipedal locomotion, 3D-LIPM and Cart-table models of the COG motion can be computed in real-time and be dynamically stable. The algorithms introduced have close-form solutions with clear geometric meaning, and therefore can be useful for developing robust real-time applications.

It was demonstrated that offline compensation of the body orientation contributes to online control, reducing high joint accelerations. As a result, a stable motion was obtained.

The stabiliser was designed as a decoupled controller. It controls the error in ZMP and Attitude positioning of the humanoid robot by the motion of the ankle and hip joints.

The humanoid robot Rh-1 provided with the developed control architecture is able to walk stably on a plain surface and absorb some external disturbances. Future work will be focussed on adding other elements to the proposed control architecture such as a foot landing control in order to correct structural and walking surface imperfections, and to reduce the mechanical landing impact on the humanoid structure, which are the essential conditions for achieving stable humanoid robot walking on irregular terrain. Also further improvements on existing mechanical, hardware and software architecture will be continued.

Acknowledgment

We would like to thank the members of the Robotics Lab of University Carlos III of Madrid for their cooperation and suggestions. We also express acknowledgements to the Humanoid team for their support. This research is supported by Comisión Interministerial de Ciencia y Tecnología (CICYT).

References

- Abraham RA, Marsden JE. 1999. Foundations of Mechanics, Second Edition. Addison-Wesley Publishing Company, Inc.: Redwood City, CA.
- Arbulú M, Balaguer C. 2007. Real-time gait planning for Rh-1 humanoid robot, using local axis gait algorithm. Paper presented at: 7th IEEE-RAS International Conference on Humanoid Robots; 2007 Nov. 29–Dec. 27; Pittsburgh, USA.
- Arbulú M, Balaguer C. 2009. Real-time gait planning for Rh-1 humanoid robot, using local axis gait algorithm. IJHR. 6(1):71–91. Print ISSN: 0219-8436. Online ISSN: 1793-6942.

- Arbulú M, Cabas LM, Staroverov P, Kaynov D, Pérez C, Balaguer C. 2006. On-line walking patterns generation for Rh-1 humanoid robot using a simple three-dimensional inverted pendulum model. Paper presented at: 9th International Conference on Climbing and Walking Robots (Clawar 2006); 2006 Sep; Brussels, Belgium.
- Arbulú M, Pardos JM, Cabas LM, Staroverov P, Kaynov D, Pérez C, Rodríguez MA, Balaguer C. 2005a. Rh-0 humanoid full size robot's control strategy based on the Lie logic technique. Paper presented at: IEEE-RAS International Conference on Humanoid Robots (Humanoids 2005); Dec 2005; Tsukuba, Japan.
- Arbulú M, Prieto F, Cabas L, Staroverov P, Kaynov D, Balaguer C. 2005b. ZMP human measure system. In: Proceedings of 8th International Conference on Climbing and Walking Robots (Clawar 2005); Sep 2005; London, United Kingdom.
- Ayyappa E. 1997. Normal human locomotion, part 1: basic concepts and terminology. *J Prosthet Orthot.* 9(1):10–17.
- Baerveldt A-J, Klang R. 1997. A low-cost and low-weight attitude estimation system for an autonomous helicopter. In: Proceedings of IEEE International Conference on Intelligent Engineering Systems. p. 391–391.
- Bicchi A, Toniatti G, Schiavi R. 2004. Safe and fast actuators for machines interacting with humans. In: Proceedings of the 1st Technical Exhibition Based Conference on Robotics and Automation, TExCRA2004; 2004 Nov 18–19; Tokyo, Japan: TEPIA.
- Cabas LM, Cabas R, Staroverov P, Arbulú M, Kaynov D, Pérez C, Balaguer C. 2006a. Challenges in the design of the humanoid robot RH-1. Paper presented at: 9th International Conference on Climbing and Walking Robots (Clawar 2006); 2006 Sep; Brussels, Belgium.
- Cabas LM, Cabas R, Staroverov P, Arbulú M, Kaynov D, Pérez C, Balaguer C. 2006b. Mechanical calculations on a humanoid robot. Paper presented at: 9th International Conference on Climbing and Walking Robots (Clawar 2006); 2006 Sep; Brussels, Belgium.
- Cabas L, de Torre S, Prieto I, Arbulú M, Balaguer C. 2004. Development of the lightweight human-size humanoid robot RH-0. Paper presented at: 7th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR 2004); 2004 Sep. 22–24; Madrid, Spain.
- Fujimoto Y, Obata S, Kawamura A. 1998. Robust biped walking with active interaction control between foot and ground. In: Proceedings of IEEE International Conference on Robotics and Automation, Leuven, Belgium. 3(16–20):2030–2035.
- Furusho J, Sano A. 1990. Sensor-based control of a nine-link BIPED. *Int J Robot Res.* 9(2):83–98.
- Gienger M, Löffler K, Pfeiffer F. 2001. Towards the design of biped jogging robot. In: Proceedings of IEEE International Conference on Robotics and Automation. 4:4140–4145.
- Hirai K, Hirose M, Hikawa Y, Takanaka T. 1998. The development of Honda humanoid robot. Paper presented at IEEE International Conference on Robotics and Automation (ICRA 1998) Leuven, Belgium. 1321–1326.
- Hirukawa H, Hattori S, Kajita S, Harada K, Kaneko K, Kanehiro F, Morisawa M, Nakaoka S. 2007. A pattern generator of humanoid robots walking on a rough terrain. In: IEEE International Conference on Robotics and Automation, Roma, Italy. Apr 10–14:2781–2187. ISSN: 1050-4729, ISBN: 1-4244-0601-3, INSPEC accession number: 9561669.
- Huang Q, Kaneko K, Yokoi K, Kajita S, Kotoku T, Koyachi N, Arai H, Imamura N, Komoriya K, Tanie K. 2000. Balance control of a biped robot combining off-line pattern with real-time modification. In: Proceedings of IEEE International Conference on Robotics and Automation Apr 2000, Budapest, Hungary. 4(14):3346–3352.
- Loeffler K, Gienger M, Pfeiffer F, Ulbrich H. 2004. Sensors and control concept of a biped robot. *IEEE T Ind Electron.* 51: 1–9.
- Löffler K, Gienger M, Pfeifer F. 2003. Sensors and control design of a dynamically stable biped robot. In: Proceedings of IEEE International Conference on Robotics and Automation; 2003; p. 484–490.
- Lohmeier S, Buschmann T, Ulbrich H, Pfeiffer F. 2006. Modular joint design for performance enhanced humanoid robot LOLA. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2006); p. 88–93.
- Kajita S, Kanehiro F, Kaneko K, Fujiwara K, Harada K, Yokoi K, Hirukawa H. 2003a. Biped walking pattern generation by using preview control of zero-moment point. In: IEEE International Conference on Robotics Automation; 2003 Sep 14–19; Taipei, Taiwan; p. 162–1626.
- Kajita S, Kaneiro F, Kaneko K, Fujiwara K, Yokoi K, Hirukawa H. 2003b. Biped walking pattern generation by a simple 3D inverted pendulum model. *Auton Robots.* 17(2):131–147.
- Kaneko K, Harada K, Kanehiro F, Miyamori G, Akachi K. 2008. Humanoid robot HRP-3. Paper presented at: IEEE-RAS International Conference on Humanoid Robots (Humanoids 2008); Nice, France; 2008 Sep 22–26.
- Kaneko K, Kanehiro F, Kajita S, Yokoyama K, Akachi K, Kawasaki T, Ota S, Isozumi T. 2002. Design of prototype humanoid robotics platform for HRP. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2002; p. 2431–2436.
- Kaynov D, Balaguer C. 2007. Industrial automation-based approach to design control system of the humanoid robot. Application to the Rh-1 humanoid robot. Paper presented at: IEEE International Symposium on Industrial Electronics (ISIE 2007); 2007 June; Vigo, Spain.
- Kaynov D, Rodríguez MA, Arbulú M, Staroverov P, Cabas LM, Balaguer C. 2005. Advanced motion control system for the humanoid robot Rh-0. Paper presented at: 8th International Conference on Climbing and Walking Robots (Clawar 2005).
- Murray RM, Li Z, Sastry SS. 1994. Mathematical introduction to robotic manipulation. Florida: CRC Press.
- Paden B. 1986. Kinematics and control robot manipulators [PhD thesis]. [Berkeley, (CA)]: Department of Electrical Engineering and Computer Sciences, University of California.
- Pardos JM, Balaguer C. 2005. Rh-0 humanoid robot bipedal locomotion and navigation using Lie groups and geometric algorithm. Paper presented at: International Conference on Intelligent Robots and Systems (IROS 2005); 2005 Aug; Edmonton, Canada.
- Park FC, Bobrow JE, Ploen SR. 1995. A Lie group formulation of robot dynamics. *Int J Robot Res.* 14(6):609–618.
- Park JH, Cho HC. 2000. An on-line trajectory modifier for the base link of biped robots to enhance locomotion stability. In: Proceedings of the IEEE ICRA 2000; p. 3353–3358.
- Raiert MH. 1986. Legged robots that balance. Cambridge: MIT Press.
- Regenstein K, Dillmann R. 2003. Design of an open hardware architecture for the humanoid robot ARMAR. In: Proceedings of IEEE International Conference on Humanoid Robots; 2003; Karlsruhe, Germany.

- Shin CL, Li YZ, Churng S, Lee TT, Cruver WA. 1990. Trajectory synthesis and physical admissibility for a biped robot during the single-support phase. In: Proceedings of IEEE International Conference on Robotics and Automation 1990; p. 1646–1652.
- Stramigioli S, Mashke B, Bidard C. 2002. On the geometry of rigid body motions: the relation between Lie groups and screws. *J Mech Eng Sci.* 216(C1):13–23.
- Torre S, Cabas LM, Arbulú M, Balaguer C. 2004. Inverse dynamics of humanoid robot by balanced mass distribution method. Paper presented at: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004); 2004 Sep; Sendai. Japan.
- Vukobratovic M, Juricic D. 1969. Contribution to the synthesis of biped gait. *IEEE Tran. Bio-Med Eng.* 16(1):1–6.
- Winter DA. 1990. Biomechanics and motor control of human movement. New York: Wiley-Interscience.
- Yamaguchi J, Soga E, Inoue S, Takanishi A. 1999. Development of a bipedal humanoid robot control method of whole body cooperative dynamic bipedal walking. Paper presented at: IEEE International Conference on Robotics and Automation (ICRA 1999); Detroit, USA.
- Yoshida E, Belousov I, Esteves C, Laumond JP. 2005. Humanoid motion planning for dynamic tasks. In: Proceedings of IEEE-RAS International Conference on Humanoid Robots (Humanoids 2005), 2005; p. 1–6.

